

Model-Reduced Gradient Based Production Optimization

Computational Analysis of POD-Based Model-Order
Reduction in the Context of Production Optimization

Dan Sava

Master of Science Thesis

Risk and Environmental Modeling
Department of Applied Mathematics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Model-Reduced Gradient Based Production Optimization

Computational Analysis of POD-Based Model-Order Reduction in the Context of Production Optimization

MASTER OF SCIENCE THESIS

For the degree of Master of Science in
Risk and Environmental Modeling
at Department of Applied Mathematics
at Delft University of Technology

Dan Sava

August 21, 2012

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Delft, The Netherlands





The work in this thesis was supported by TNO. Their cooperation is hereby gratefully acknowledged.



All rights reserved.

Copyright © Applied Mathematics Department

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Delft, The Netherlands

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
APPLIED MATHEMATICS

The undersigned hereby certify that they have read and recommend to the Faculty
of Electrical Engineering, Mathematics and Computer Science for acceptance a
thesis entitled

MODEL-REDUCED GRADIENT BASED PRODUCTION OPTIMIZATION

by

DAN SAVA

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE.

Dated: August 21, 2012

Supervisor:

Dr. Ir. Remus Hanea

Readers:

Prof. Dr. Ir. A.W. Heemink

Dr. Ir. O.Leeuwenburgh

Prof. Dr. Ir. J.D. Jansen

MSc. S.P. Szklarz

Abstract

In the context of reservoir engineering, determining the ideal strategy, in order to get an optimized oil production, is one of the most sought after objectives. Taking into consideration the fact that we are currently living in a world dominated by the pursuit of fast and accurate results, what we are trying to do, is to come up with the most accurate production strategy in the shortest amount of time. A standard production optimization implementation requires numerous model runs during the course of the optimization process. A practical reservoir model contains thousands, if not millions of grid blocks, that in turn contain multiple parameters. Running this kind of model several times results in quite a time consuming endeavour. Then the main problem that this study addresses is: how can the computational cost for the optimization process be reduced in such a way that it still provides sufficiently accurate optimization results. One way to tackle this problem is by reduced-order modelling procedures, seen as an alternative to the classical gradient based approach, in which the adjoint of the tangent linear approximation of the forward model is replaced by the adjoint of a linear approximation of a reduced forward model. Proper orthogonal decomposition is used to compute the reduced model, by using the reduced adjoint, the gradient of the objective function can be approximated and the maximization problem can be solved in a reduced space. This work researches the viability of performing production optimization based on model-order reduction, by comparing several aspects of the implementation with similar aspects of Ensemble Optimization, a already well known way of solving the production optimization problem, that computes the gradient information from an ensemble of model realizations. The conclusion on whether this method is suited for practical applications is drawn after the analysis of aspects such as computational cost, accuracy of the results and implementation difficulty, which were compared with EnOpt first on a simple 2D reservoir model and then on a more complex 3D reservoir model.

Table of Contents

Acknowledgements	ix
1 Introduction	1
1-1 Literature Review	2
1-1-1 Parametrization	4
1-2 Chapter Outline	5
2 Closed Loop Reservoir Management	7
2-1 Data Assimilation	9
2-2 Production Optimization	10
3 Reservoir Flow Model	13
3-1 Reservoir Geometry and Properties	13
3-2 Phase Behaviour	14
3-3 Flow Modelling in Porous Medium	15
3-4 State space representation of the reservoir flow model	20
3-4-1 The discrete-time state space representation of the reservoir flow model	20
3-5 Adjoint Model for the Reservoir Model	21
3-5-1 Difficulties implementing an adjoint model	23
4 Reduced Order modelling	25
4-1 Motivation	25
4-2 Projection-Based Reduced Order modelling Methods	26
4-2-1 Snapshots	28
4-2-2 Projection matrices	29
4-2-3 Reduced order model	31
4-2-4 POD method comments	32

5	Model-Reduced Gradient Based Production Optimization	33
5-1	Introduction	34
5-2	Reparametrization	34
5-3	Gradient-Based Model-Reduced Production Optimization	35
5-3-1	Initialization	36
5-3-2	Collection of the snapshots and pattern selection	36
5-3-3	Building the reduced-order linear model	37
5-3-4	Improving parameters in the reduced space	38
5-3-5	Improving controls for the full-order model	39
5-3-6	Computational complexity	40
5-3-7	Additional issues	41
5-4	Ensemble-Based Optimization	42
5-4-1	Ensemble optimization	42
5-4-2	Gradient optimization	44
5-4-3	Implementation of ensemble-based optimization	47
6	Numerical Results	49
6-1	Matlab Reservoir Simulation Toolbox (MRST)	49
6-2	Numerical Results 2D Case	50
6-2-1	Production optimization settings	53
6-2-2	Production optimization results	53
6-2-3	Conclusions	61
6-3	Numerical Results 3D Study Case	62
6-3-1	Conclusions	70
6-4	Summary	71
7	Conclusions	73
7-1	Conclusions	73
7-2	Further Research and Recommendations	75
A	Appendix	77
A-1	Figure Appendix Section	77
	Bibliography	81

List of Figures

2-1	Closed-loop model-based reservoir management (Jansen et al 2009). . .	8
2-2	Optimization Tree (Optimization Technology Center, 2008)	11
2-3	Important elements of the reservoir simulation	12
3-1	Reservoir Grid Block representation	14
3-2	Corey type relative permeabilities.	19
4-1	Simulation is the third big discipline.	26
5-1	Model-reduced gradient-based production optimization	40
5-2	Expected Value of the Objective Function	43
5-3	The flowchart for the Ensemble-Based Optimization	46
6-1	The reservoir Grid used in this section for the numerical experiments. The red squares correspond to the producers and the blue squares cor- respond to the injectors	51
6-2	Relative permeability curves.	52
6-3	NPV for the case when the optimization process is started using different initial controls	54
6-4	Water saturation distributions at six month time interval	55
6-5	MorOpt Parameter Sensitivity	56
6-6	2D Comparison of Optimization Results	58
6-7	2D Optimized Control Strategy	59
6-8	Cumulative Water Production for each Production Well	59
6-9	Cumulative Water Production	60
6-10	Cumulative Oil Production for each Production Well	60
6-11	Cumulative Oil Production	61

6-12	Reservoir permeability field with well locations. Injectors are indicated in black, producers in red	63
6-13	Comparative Optimization Results	65
6-14	EnOpt two iterative methods	66
6-15	Comparative Production Strategy	67
6-16	Optimized Water Injection Strategy	68
6-17	Water Production Under Optimized Controls	68
6-18	Oil Production Under Optimized Controls	69
A-1	NPV evolution: MorOpt reparametrize controls # Samples 25 relative importance 95% (blue line) MorOpt reparametrize controls # Samples 50 relative importance 95% (red line) MorOpt reparametrize controls # Samples 50 relative importance 75% (gree line) EnOpt # Samples 50 trust region (black line) EnOpt # Samples 50 line search (magenta line)	77
A-2	Production strategy: MorOpt reparametrize controls # Samples 25 relative importance 95% (green line) EnOpt # Samples 50 trust region (blue line)	78
A-3	NPV evolution	78
A-4	Production strategy Figure (A-3)	78
A-5	2D MorOpt Parameter Sensitivity Analysis	79
A-6	3D Production Strategy Comparison	79
A-7	2D Cumulative Oil and Water Production	80
A-8	3D MorOpt NPV Comparison	80

List of Tables

6-1	Optimization results for different background controls	55
6-2	Optimization results for different parameter values	56
6-3	Comparison of Optimization Results 2D case	57
6-4	Comparison of Total Liquid Volumes for 2D case	61
6-5	Comparison of Optimization Results 3D Case	66
6-6	Comparison of Total Liquid Volumes for 3D case	70

Acknowledgements

I would like to thank the people that put together the Risk and Environmental Modelling track of the Applied Mathematics TUDelft MSc Program. Thank you for offering me the possibility to further my understanding of the physical world and how it can be described using simple mathematical models.

I would also like to thank my advisor, Remus Hanea, for the time he spent with me this year, for introducing me to the world of Data Assimilation and Reservoir Engineering. Learning from him, especially learning how to ask the right questions, has proved an invaluable experience. His guidance and encouragement made this work possible.

I am also very grateful to my thesis committee members, A.W. Heemink and O.Leeuwenburgh for the fruitful discussions, which substantially improved my understanding of the problem and challenges that I had to face during the construction of this thesis.

Special thanks to my group-mates, including but not limited to Aurelius Zilko, Diana Lucatero, Dana Stuparu, Liuxin Si, Qian Feng, Donghua Zhang, and Nikita Moriakov, who were always available for a friendly discussion either technical, or totally non-technical.

Also, I would like to thank S.P. Szklarz and M.P. Kaleta for a very interesting discussion on Model Order Reduction.

Finally, I give thanks to my family, G. Georgescu and M. Constantinescu for their simple everyday understanding, patience, and support.

Chapter 1

Introduction

About ninety million years ago, in a period call the late Cretaceous when dinosaurs still roamed the earth it was a period of extreme global warming. As continents drifted apart, ocean water flooded the rifts that were forming and algae were thriving in the newly formed seas. The algae died and fell in the bottom of the rifts and over a long period of time sediments, washed by rivers, covered the organic material. As the sediment layer grew so did the pressure and heat until a chemical reaction transformed the organic material into hydrocarbon fossil fuel. Our modern way of life is entirely dependent on these fossil fuel, but most of the population in developed countries takes this for granted. Since 1860 geologist have discovered over two trillion barrels of oil and since then the world has used approximately half. At first the oil was easy to find and cheap to extract, but for a oil well is only a matter of time until the production values start to decline. This means that the reservoir production has peaked and from that point on the production is permanently declining.

Looking at the rate of discovery for new oil fields we can see a clear and constant decline. In the past, for every six barrel of oil discovered we consumed about a barrel, now we have reached the point when for every barrel of new oil discovered we consume three to four barrels of oil. Combining this with the fact that modern society is not yet ready to satisfy its energy needs by using alternative energy sources like solar, wind nuclear or geothermal. We come to the logical conclusion that new optimized production strategies need to be developed and implemented in order to stop or keep current oil reservoirs from peaking in production to early. Also we need to produce as much oil as possible, until a solution to the energy need is found. One such production strategy is known as water flooding, which is the secondary recovery phase in the reservoir life cycle. Water is injected through water injection well in the reservoir in order to re-pressurize the reservoir and to produce more oil. The other parts in the reservoir life cycle are, the primary recovery phase, which represents the initial reservoir production under the natural pressure difference between inside

the reservoir and the well pressure, this phase usually takes several years and yields around 20-30% recovery. The tertiary recovery phase or enhance oil recovery phase, gas, chemicals or steam are injected to alter the fluid's properties and wash the rock. For example during steam recovery steam is injected to alter the viscosity of the oil and make it flow more easily.

The optimization of the water flooding stage is the main objective of this thesis. The optimization is done with the help of a model order reduction optimization method (MorOpt). Computer simulations are now performed routinely for many physical, chemical and other processes, and virtual design environments have been set up for a variety of problem classes in order to ease the work of designers and engineers. In this way, new products can be designed faster, more reliably, and without having to make costly prototypes. Originally, Reduced Order Modeling (ROM) or Model Order Reduction (MOR) was developed in the area of systems and control theory, which studies properties of dynamical systems in application for reducing their complexity, while preserving their input-output behavior as much as possible. In the oil and gas industry ROM techniques provides a computationally efficient prototyping tool that replaces large scale models with approximate reduced order models, that are capable of capturing the essential dynamical behaviour and also preserve critical properties of the large models. A good model reduction technique will try to focus on the necessary information and ignore irrelevant information that comes from the high order model. One of the main reasons for using model reduction techniques is computational speed-up of simulation time. As a consequence of this simulation speed-up, a simulation might finish in hours instead of days which is customary for some reservoir cases, then re-evaluation of method which prior were considered too computationally expensive can be done or improve results by allowing for more simulations.

1-1 Literature Review

The aim is to optimize an objective function with respect to the control values, during the water flooding operations. There are several ways to accomplish this goal. One is the simultaneous perturbation stochastic approximation (SPSA) which is an alternative gradient based algorithm where an approximated gradient is calculated using a stochastic perturbation of all parameters together, which was proposed for use in the recovery optimization by Wang et al. (2009). Another way to realize the goals of production optimization is by employing a method, known as "ensemble optimization", that uses an ensemble of control variables and computes the cross-covariance between the control variables and the objective function, which can successfully be used as an approximation of the gradient. It was first proposed for recovery optimization by Lorentzen et al. (2006) where it was shown that the limitations of optimal control could be avoided, given the fact that adjoint equations are not needed and the model equations are treated as a black box. This is a powerful technique, and was refined by Chen et al (2009), where it was

shown that it can be an integral part of closed-loop production management. A method that could be considered an alternative to these two previously mentioned methods is the model-reduced production optimization method. It was proposed by Vermeulen and Heemink (2006) and avoids implementing the adjoint model of the tangent linear approximation of the original nonlinear model. The method is based on the fact that the adjoint of the tangent linear approximation of the original forward model is replaced by the adjoint of a reduced linear forward model, using this reduced adjoint the gradient of the objective function is approximated and the maximization problem is solved in the reduced space. In this approach, the Proper Orthogonal Decomposition (POD) method is used to obtain an approximate low order version of the tangent linear model. In the POD method, introduced by Karhunen and Loeve (1964), the subspace on which the projection is done is constructed by processing data obtained from numerical simulation of the full order model. The information obtained from the numerical simulations are expected to contain significant insight on the dynamic behaviour of the system. The high-dimension equations are projected onto the low-dimensional subspace resulting in a low dimensional model. Vermeulen and Heemink (2006) use the POD method in a somewhat different manner when doing the state space reduction applied for ground water flow problems. Analysing the results, a significant computational advantage, can be observed. Later a similar method was used for production optimization J. van Doren (2006) and history matching M.P. Kaleta (2011).

Since the reservoir models are described by a nonlinear system of equations, it becomes more difficult to solve the optimization problem, than in the case of a linear system. The difficulties appear in the implementation of optimization methods, as well as in the chance of finding the global optimum. Well known methods to solve this problems are iterative gradient-based optimization techniques, where the gradient is calculated by the so called adjoint method. The problem for these types of methods is the implementation, this is due to the complex form of the reservoir model equations. This results in a tremendous implementation effort to obtain the adjoint model and the sensitivities of the model with respect to the controls. Usually in reservoir models, the Jacobian matrices of the system are available because they are used in a Newton-Raphson time step iteration during reservoir simulation. Even so, the implementation of the adjoint equations, required in gradient calculations is a considerable programming effort, which also requires access to the simulation code. This is not something that can be easily obtained for commercial reservoir simulators, it also implies that there is a strong need for gradient-based, but adjoint free optimization methods. MorOpt, the optimization method studied for this work, fulfils exactly these needs, and represents a gradient-based, adjoint free optimization method that works with the reservoir simulator as a black box. The approach studied obtains a reduce-order approximation of the tangent linear model by projecting approximate derivatives of the full order model. Then due to the linear characteristic of the reduce-order model, the corresponding adjoint model is easy to obtain. The gradient of the objective function is approximated and the maximization problem is solved, in the reduced space. The procedure can be iterated for the updated controls if necessary, to look for farther improvements of the control vector. Also from the numerical experiments results, we will see that

MorOpt proves to be a fast optimization method, with respect to the number of times the reservoir simulator is required to run in order to get a final optimized control vector. Also it is considerably less difficult to implement than a classical gradient based method that requires the presence of the adjoint.

1-1-1 Parametrization

When dealing with a gradient based production optimization algorithm, the gradient of the objective function with respect to the controls is required. Using the adjoint based method it can be calculated by one solution of the reservoir model and one solution of the corresponding adjoint model independently of the number of controls. For production optimization the control parameters represent, particular injection rates at specific times during the life cycle of the reservoir. The number of controls is computed as the product of control steps and number of wells, this means it grows with the scale of the problem. The object of parametrization is to try and express the model control parameters using a smaller number of control parameters. This is done while preserving the important geological features in the new parametrization. Further two parametrization methods are briefly discussed for potential application in reparametrizing the control space of the production optimization problem.

Discrete cosine transform

This method originates in the image processing field and is seldom used as an image-compression standard. It was applied by Jafarpourt and McLaughlin (2009) for parametrization of a permeability field, in a history matching problem. Using this approach, from a production optimization problem, one would express a control vector as a linear combination of predefined basis functions weighted by uncorrelated random coefficients. The basis function are defined as cosine functions. The total number of basis function is equal with the number of control values. If the control vector was known only the basis function with the highest coefficients in the expansion would be retained, which in turn would reduce the number of control parameters. Dealing with the fact that the optimal control vector, is unknown, it is not possible to identify the most relevant basis functions, by ordering the coefficients. This poses problems when implementing this parametrization technique for production optimization problems.

Karhunen-Loeve expansion

The Karhunen-Loeve expansion (Karhunen (1946), Loeve (1946)), is a powerful parametrization approach that has been very well studied for history matching. It has been applied for permeability and porosity parametrization by Gavalas et al (1976), Reynolds et al. (1996). Using this approach in production optimization, each control parameter can be expressed as a linear combination of deterministic basis functions weighted by uncorrelated random coefficients. The basis functions

are defined by the eigenvectors of a specified control covariance matrix and are obtained by solving a eigenvalue problem. In order to reduce the eigenvalue problem the covariance matrix can be approximated by the sample covariance. In such a case the parametrization depends on the given data. Usually only the eigenvectors that correspond to the largest eigenvalues are used. The eigenvectors that are not used correspond to the shortest correlation lengths present in the sample.

1-2 Chapter Outline

The second Chapter of this thesis presents the closed loop reservoir management, as an essential part in increasing the maximum output of an oil reservoir. Chapter 3 describes the reservoir flow model used when building the mathematical models that simulates what is happening deep inside the reservoir. Chapter 4 focuses on Model Order Reduction, describes the motivation and theoretical part behind the method. The implementation of the method is described in Chapter 5, which also contains the description of Ensemble Optimization another gradient-based optimization method. Chapter 6 presents results obtained from numerical simulations done on a simple 2D grid model and a more complex 3D grid model.

Closed Loop Reservoir Management

When dealing with oil reservoirs, the initial geological model is a physical representation of the underground reservoir. This representation is based on accumulated knowledge from geology, exploration wells and seismic data. All of these are combined to form a numerical model which is typically very complex and detailed, containing sometimes millions of grid blocks. For each grid block of the model, most of the time two parameters are defined, permeability and porosity. These two parameters are essential for the next phase of the construction of the numerical model for the reservoir, which is describing the flow through the reservoirs and into the production wells. This description is done with the help of partial differential equations, which are based on the mass balance law and Darcy's law combined with initial and boundary conditions. These equations form a complex set of coupled equations. Using numerical methods, model variables, such as pressures and saturations of phases or accumulations of components, can be calculated at discretized time on the space domain.

Using the prior model, predictions regarding future performance of the reservoir can be made. If predictions appear to be economically profitable, the reservoir goes in the field development stage. During this phase the main objective is to maximize the expected profit. For this maximization to be possible, a numerical model is used, which is expected to mimic as close as possible the processes that are taking place in the reservoir during the production phase. In the best case scenario the numerical model fully describes the behaviour of the reservoir system, in which case an optimal operational strategy can be found by maximizing an objective function representing the net present value (N.P.V.).

Most unfortunate is that a numerical model can only describe the reality in a broad way. This may be caused by several things, from which worth mentioning are the following: the processes that are occurring in the reservoir are far too complex

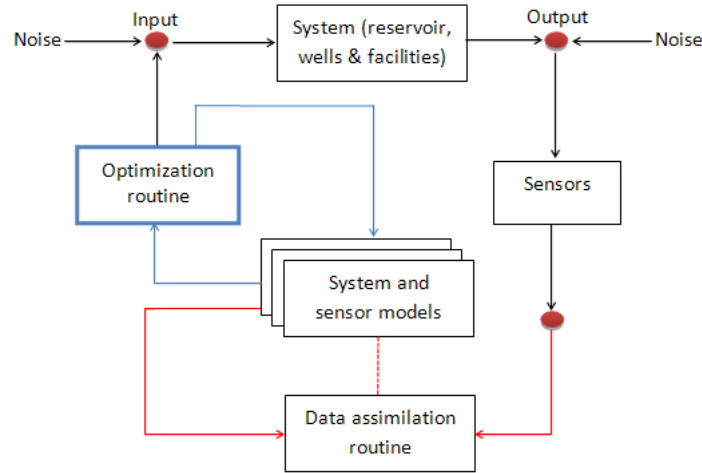


Figure 2-1: Closed-loop model-based reservoir management (Jansen et al 2009).

and the fact that they are happening deep underground where simple observations cannot be performed only worsens our ability to come up with accurate models for them. This unreliable understanding of the processes that are taking place in the reservoir, to a certain degree, can be implemented as model error. One other thing that can cause a numerical model to provide an inaccurate representation of reality comes from the fact that rock properties, such as porosity and permeability, appear to be poorly known. These rock properties map to inaccurate flow equations. One way to deal with these errors is to implement them in the model as uncertain parameters. Ignoring these uncertainties and just simply assuming that the properties are correct may lead to erroneous predictions and wrong decisions. During the production phase, data from the production wells is collected and compared with the predicted production from the numerical model. A difference is always noticed between these two elements, because as previously mentioned the model is just a crude approximation of reality. This means that the numerical model needs constant adjustment so that it can be closer and closer to reality. Measuring the well data together with the numerical simulations from the model can be used to improve the model and increase the recovery from an oil field.

In recent years the development of closed-loop model-based reservoir management, in the oil industry, has brought together the procedures for model updating and production optimization. It has integrated the two concepts into one structured approach to provide real time reservoir management Jansen et al (2009). Figure (2-1), presents the main ideas behind the closed-loop reservoir management through a flow chart representation. The system box represents the reality, in the form of the reservoir with wells, facilities and sensors. The system is controlled by a number of inputs which are located in the wells and also influence the flooding process. The input that reaches the system is known to a limited extent due to the unknown reservoir properties which are indicated by noise in the Figure (2-1). Production data is

monitored via an array of sensors located in the wells, or by additional seismic data gathered all over the reservoir. The measured data is assumed to be contaminated with measurement errors, again these errors are represented by noise in the Figure (2-1). The system and sensors models box represents the model equations that are supposed to mimic reality and provide a forecast of the production rate. The multiple boxes represent the possible geological models of a given reservoir. These models are created after a detailed analysis of the available data and then they are up-scaled. The figure contains two loops, one red and one blue. The loops represent two processes, the "red loop" represents the data assimilation process while the "blue loop" represents the production optimization process. Compared to the standard approach of performing either production optimization or data assimilation, what the closed-loop model-based reservoir management does, is that by combining data assimilation with production optimization it improves overall oil production. The idea is that when production data is available, this data, via the data assimilation loop is used to update the model parameters, resulting in a more accurate model. The assimilation is then followed by an optimization loop, "blue loop", that uses the new updated model or models to provide the updated optimal strategy for future production. The procedure is supposed to be iterated every time new data is available.

2-1 Data Assimilation

Data assimilation combines information from the numerical model and the information present in the observations, as well as taking into account the uncertainties that come from the observations and the model to provide more realistic results. Applications of data assimilation can be found in various fields such as, groundwater flow, meteorology, oceanography, and petroleum engineering. Data assimilation can be used in any field where the existing numerical models do not provide an accurate description of reality. This happens because physical properties of the system are hard to observe, understand and ultimately describe in a numerical way.

In the petroleum engineering data assimilation is regarded as history matching and its main focus is to adjust the model parameters. These model parameters are plagued with uncertainty, they usually refer to rock properties such as porosity and permeability, but might also refer to fluid properties such as viscosity and relative permeability. An initial characterization of the reservoir is based on wellbore and logging data that are interpolated to give regional descriptions of uncertain geological properties. The uncertainty in the those parameters directly translates to model uncertainty. History matching improves the geological model by using subsurface and down-hole production data, represented by flow rates and bottom-hole pressures. Looking at history matching as an optimization problem, what it does is it minimizes an objective function that represents the differences between the model prediction and the observed production data.

Over the last few decades, in which history matching has been investigated, several algorithms that tackle this problem have been developed. They can be categorized into four groups: gradient-based algorithms, Kalman filter algorithms, streamline-based algorithms and stochastic algorithms. All these algorithms have positive and negative parts to them regarding aspects such as: computational efficiency, implementation complexity, assumptions on the flow model, the capability to support complex geological constraints and the sensitivity to local optima. Two of the four types of the algorithms used in history matching will be described, gradient based algorithms and Kalman filter based algorithms. These were chosen because they manage to provide the best accuracy versus computational cost ratio.

The gradient-based history matching algorithms represent a particularly efficient class of history matching methods, that involves calculating the gradients of the objective function with respect to the model parameters, by solving an adjoint or co-state problem. So in the adjoint approach, the history matching problem is treated as an optimal control problem, in which the control variables are the unknown model parameters and the objective function is minimized subject to the constraint that the state variables obey the prescribed reservoir model. The adjoint approach in most cases is very efficient from a computational stand point. A single simulation of the reservoir model is required for one gradient calculation when the adjoint is available.

The Kalman filter method was proposed by Kalman in 1960, and it represents a recursive method that estimates the state of a linear dynamical system from a series of noisy measurements. It is assumed that both the initial state vector of the system and the measurements vector are riddled with uncertainties. Using the model equations, a prediction of the state is computed in what is called the forecast step, this is done up to the moment new observations are available. In the analysis step, an estimate of the state and its uncertainty is calculated by taking into the account the available observations. The correction is done using the Kalman gain matrix which is a matrix of weights that represent the combination of information from the uncertain observation data and the information from the uncertain model state. The method assumes an independent Gaussian world as the reality. This means that the measurement noise is independent in time, there is no correlation between model noise and measurement noise and that the measurement noise has Gaussian distribution. Under all these assumptions the conditional probability is Gaussian and is fully characterized by the first two statistics, the mean and the covariance.

2-2 Production Optimization

During the production optimization phase of the reservoir management process, one tries to identify the best position of wells and a configuration of controls that would provide a maximized production from the reservoir. Describing the process as an optimization problem, what we are trying to do, is to maximize an objective function, in this case also known as the Net Present Value or Total Hydrocarbon

Recovery. Due to the fact that the economical objective that is improved involves evaluations of future achievements, it requires a reservoir simulation model for prediction. It is usually referred to as a model based optimization technique. Since in most cases the underlying geological reservoir model is far from homogeneous, the fluids flow with different velocities in different parts of the reservoir. This means that, during water flooding, the injected fluid will move much faster through paths which are highly permeable and if the high permeable region is connected with a production well it will reach it very fast. The oil located outside this region is left in the reservoir, resulting in a poor reservoir flooding. Continuing the injection will result in the production of water instead of oil. This is known as water breakthrough, something that is best avoided. To avoid early water breakthrough, one needs to find an operation strategy that will sweep the oil uniformly towards the production wells. This is usually done by changing the injection production controls in the wells. To test whether one such strategy is efficient it should result in higher recovery compared to the reactive production strategy, in which no action is taken until significant changes are observed in the production wells.

Depending on the type of optimization problem that needs solving, some optimization techniques perform better than others with respect to robustness and efficiency. Generally, a trade off between robustness and efficiency is expected, deterministic search algorithms usually are more robust than global search algorithm, while the latter, as their name suggest, have a better chance of finding a global optimum. Looking at a typical classification of optimization problems, mainly divided into continuous and discrete optimization problems it easy to observe:

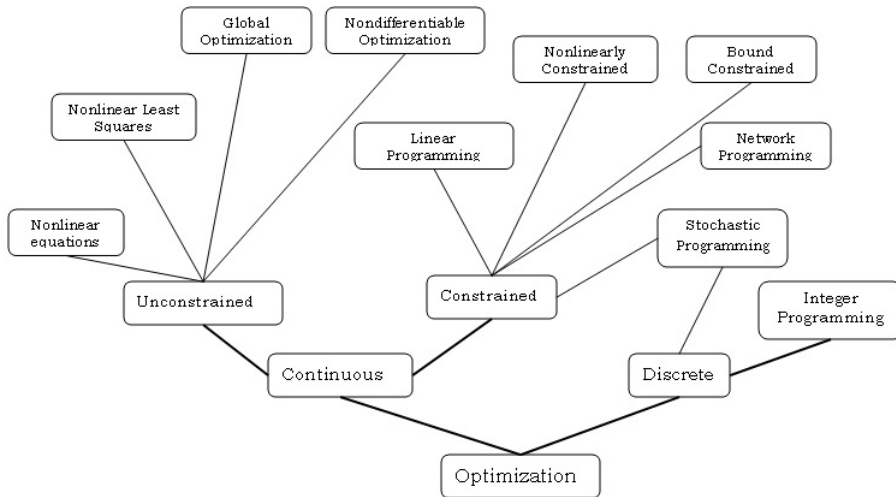


Figure 2-2: Optimization Tree (Optimization Technology Center, 2008)

To check whether the chosen injection rates optimize the production profit the Net Present Value is evaluated. The NPV is defined as:

$$J = \int_0^{t_N} \phi(t) dt \quad (2-1)$$

Where t_N stands for the total simulation time and $\phi(t)$ represents a linear combination of well rates combined with monetary factors. In the case we are dealing just with injection rates then it looks like this:

$$\phi(t) = \sum_p (r_o^p(t)q_o^p(t) - r_w^p(t)q_w^p(t)) - \sum_i r_w^i(t)q_w^i(t), \quad (2-2)$$

The first summation of equation (2-2) is over the production wells, while the second one is over the injection wells. The rates q_o^p, q_w^p stand for the volume rates of oil and water at reservoir conditions in the production well p at time t , q_w^i stands for the injection rate of water in the injector i at time t . The weight factors, $r_o^p(t)$ and $r_w^p(t)$ stand for the monetary of other type of value that is assigned to a one unit volume of water or oil at time t . The weighting factor $r_w^i(t)$ stands for the value assign to the unite volume of injected water in the injection well i at time t . For each produced fluid the annual interest rate is take into account by defining $r_l^p(t)$ as:

$$r_l^p(t) = r_l^p(t_0)f(t)(1 + b_l)^{-(t-t_0)/year} \quad (2-3)$$

Where $l \in \{o, w\}$, w stands for water and o stands for oil and $r_l^p(t)$ is the unit price of component l at reference time t_0 , $f(t)$ is the price factor for component l at time t , b_l is the annal discount factor for component l and t_0 is the discount reference time.

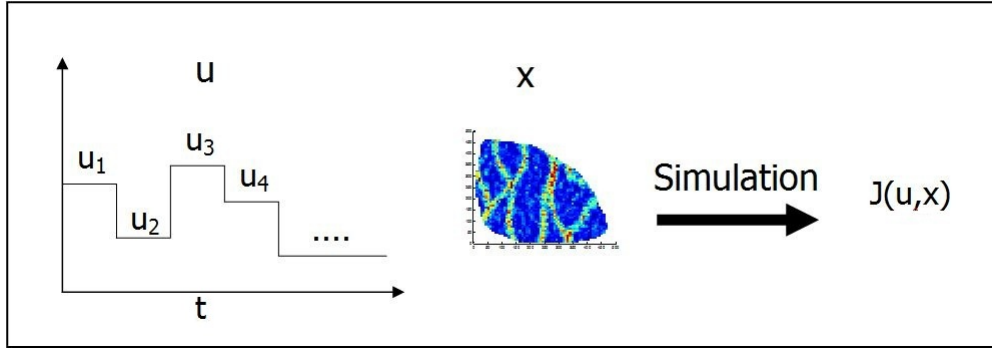


Figure 2-3: Important elements of the reservoir simulation

The aim of production optimization is to adjust the values of the control vector in such a way that the resulting objective function reaches its maximum. In a typical application, for the purpose of preventing unphysical values the control vector is restricted between specified lower and upper bounds. The the optimization problem can be written as:

$$\begin{aligned} \max_u J(u, x) &= \sum_{i=0}^{N-1} J_{i+1}(u_{i+1}, x_{i+1}) \\ \text{subject to } &u_{low} \leq u \leq u_{up} \end{aligned} \quad (2-4)$$

Reservoir Flow Model

Predictions of reservoir performance are essential when business decisions regarding field exploration need to be taken. In order to predict reservoir performance a simulation model is created. This model describes the physical processes taking place in a reservoir and is solved numerically. Typically, commercial software is used, in the case of this study MRST-Sintef, an open source reservoir simulator is used. Reservoir flow simulation usually is performed in two stages: first, one or more geological models are created by geologist and geophysicists which aim to provide a static description of the reservoir; second, reservoir simulation models are created by reservoir engineers using information about reservoir geometry, and fluid properties that are present in the reservoir. These dynamical reservoir models can be used to simulate reservoir performance under different processes involved in oil recovery, such as: water flooding, steam injection, polymers or chemical flooding.

3-1 Reservoir Geometry and Properties

Depending on the study case, the domain of the model can either be three dimensional or two dimensional. Reservoirs with two dimensional geometry are used for testing hypothesis before moving to a more realistic three dimensional model. The typical grid that is used to model reservoir geometry is a Cartesian grid, where the grid blocks are defined as small boxes. There are other types of grids that can be used in place of Cartesian grids, such as: triangular and tetrahedral grids, 2D Voronoi grids, extrusions of areal grids to 2.5D volumetric grid. All these types of grids are supported by the reservoir simulator chosen for the numerical experiments. Figure 3-1 is an example of a representation of reservoir geometry using a Cartesian grid, this type of representation was used for all the numerical experiments. To guarantee an accurate solution of the simulation, the grid blocks should be small enough to accurately represent the reservoir heterogeneities, but large enough to assure that the computational time required for the simulation stays within acceptable limits. For each grid block of the model the absolute permeabilities, (pressure

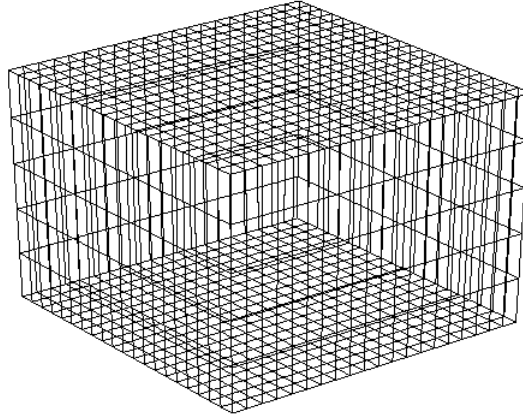


Figure 3-1: Reservoir Grid Block representation

dependent) porosities need to be specified. For reservoirs that contain fractures optional permeability compaction data needs to be specified too. For the purpose of presenting a simplistic flow model in this chapter a model with a two dimensional Cartesian grid is used, with no flow boundary conditions, no fractured structure, no aquifers and no faults are present.

3-2 Phase Behaviour

The reservoir is considered to be a porous medium, in which the pores are filled with hydrocarbons and water. The hydrocarbons are a mixture of large number of chemical components with different densities at different concentrations. Although the best case scenario would be construct a full multi component model this is usually impossible due to the increase in computational time. Therefore, simplifications of the reservoir fluid description are required so that a computationally acceptable model can be obtained. The most simple approach is to model a phase by considering it is comprised out of just one chemical component. Usually this assumption is used only for the water phase, but is not valid for the oil phase. Since the oil is mainly comprised out of two hydrocarbon components, heavy and light hydrocarbon, it is common practice to assume that the hydrocarbon mixture contains two pseudo components. The heavy component is defined as the part of the reservoir fluid which, when brought to the surface will become the stock tank oil and the light hydrocarbon component on the surface becomes the stock tank gas. All these previously mentioned assumption lead to something that is well known in the industry, called the black oil description. The distribution of the components over the phases are specified using a semi-empirical rule. After the phase distribution is set, different properties of each phase require to be specified, such as: viscosity $\mu[Pa \cdot s]$, density $\rho[\frac{kg}{m^3}]$ and phase compressibility. In our study case, the reservoir simulator used for the numerical simulations, does not support compressible fluid,

meaning that the phase compressibility will always be zero. Given the fact that there is no direct access to the reservoir, these properties are not known and cannot be measured easily, so they are derived by fitting the parameters to results of well tests and laboratory test. Since the density and viscosity depend on pressure, temperature or fluids composition, these dependencies need to be modelled too. In the case of liquids like water or oil, the density and viscosity depend weakly on the pressure, while for the gas or vaporous oil, they depend strongly on pressure and relatively weakly on the fluid composition. The viscosity of oil and gas in most cases shows a strong dependence on temperature. The simplistic flow model that is considered in this thesis assumes the presence of oil and water phase and constant viscosity, density and porosity.

3-3 Flow Modelling in Porous Medium

The way common reservoir simulators work, is that at each time step of a simulation, pressure, saturation and temperature are determined, these three are called primary variables or state variables. For the purpose of this study isothermal conditions within the reservoir are assumed. This means constant temperature and therefore temperature stops being one of the primary variables. The derivation of the flow equations in porous media is based on two very important laws, the law of conservation of mass and Darcy's law. For the purpose of our study it is assumed the reservoir acts as a partially closed system, there is now flow across the boundaries of the reservoir, i.e. the reservoir is surrounded by impermeable rock, and the only exchange of energy (flow of liquid) occurs through the wells. This characterization results in the following partial differential equations for phase $l = o, w$, where o stands for oil and w stands for water:

$$\nabla \cdot (\alpha \rho_l v_l) + \alpha \frac{\partial(\rho_l \phi s_l)}{\partial t} - \alpha \rho_l q_l \quad (3-1)$$

The first term in equation (3-1) is called the transmissibility term and describes the difference in the mass of phase l that flows with the velocity v_l in and out of a unit volume per unit time. The second term, which is also called the accumulation term, describes the accumulation of the mass of phase l per unit time and volume. The third term, called the source/sink term, describes the mass added or extracted from the reservoir, through wells per unit time and volume. α stands for a geometric factor, which accounts for the translation of the 1D or 2D model into the 3D. Now looking at the individual components that construct the three terms of equation (3-1) we have: s_l represents the saturation of phase l , t stands for time, and usually is measured in seconds, ρ_l is used to denote the density of phase l and is measured in $\frac{kg}{m^3}$, ϕ , represents the porosity of the rock, q_l represents the well rate per unit volume of phase l , measured in $\frac{1}{s}$, last but not least, the velocity of phase l in the porous medium, v_l which is given by Darcy's law:

$$v_l = -\frac{k_r l}{\mu_l} K(\nabla p_l - \rho_l g \nabla d) \quad (3-2)$$

where p_l , measured in pascals Pa denotes the pressure, μ measured in $Pa \cdot s$, denotes the viscosity and k_{rl} denotes the relative permeability of phase l . g is the gravitational acceleration and d measured in meters represents the depth at which the reservoir is situated. K is the absolute permeability tensor, which is constructed as a 3×3 matrix. In order to get a more simplified description the coordinate system can be aligned with geological layering in the reservoir such that the permeability tensor becomes a diagonal matrix :

1. For the 1D case: $K = k$
2. For the 2D case: $K = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}$
3. For the 3D case: $K = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}$

Even though it almost never really occurs it is assumed that permeability is isotropic, meaning that the permeability in the x direction is equal with the permeability of the y direction ($k_x = k_y$). For the purpose of simplifying the presented reservoir model it is assumed that the isotropic permeability field is horizontal ($k_x = k_y = k$), this is not a realistic assumption and is taken just for purpose of simplifying the theoretical case. Now by substituting the Darcy's velocity defined in equation (3-2) instead of the phase velocity in equation (3-1) the following system of partial differential equations is obtained:

$$\nabla \cdot \left(-\alpha \rho_l \frac{k_{rl}}{\mu_l} k (\nabla p_l - \rho_l g \nabla d) \right) + \alpha \frac{\partial(\rho_l \phi s_l)}{\partial t} - \alpha \rho_l q_l = 0 \quad (3-3)$$

This is a system of two equations and four unknowns, which cannot be solved without the help of additional information that is provided in the form of closed equations.

$$s_o + s_w = 1 \quad (3-4)$$

and

$$p_o - p_w = p_c(s_w) \quad (3-5)$$

where $p_c(s_w)$, measured in Pa represents the capillary pressure, whose physical meaning shall be discussed further on. In the simplistic model we are working with, it is further assumed that density depends on pressure; therefore the accumulation term becomes:

$$\frac{\partial(\rho \phi s_l)}{\partial t} = \frac{\partial \rho_l}{\partial p_l} \frac{\partial p_l}{\partial t} \phi s_l + \frac{\partial \phi}{\partial p_l} \frac{\partial p_l}{\partial t} \rho_l s_l + \frac{\partial s_l}{\partial t} \rho_l \phi. \quad (3-6)$$

In the case of liquids (oil or water), the density is just weakly dependent on pressure and its sensitivity can be expressed by using the compressibility of the phase, which for a constant temperature is defined in the following way:

$$c_l(p_l) := \frac{1}{\rho_l} \frac{\partial \rho_l}{\partial p_l} \quad (3-7)$$

Further more if the capillary pressure effects are very small, the oil pressure is approximately equal to the water pressure and the compressibility of the water can be expressed as a function of the oil pressure.

$$c_w(p_o) := \frac{1}{\rho_w} \frac{\partial \rho_w}{\partial p_o} \quad (3-8)$$

The compressibility effect appears not just for liquids oil and water, it can be defined also for the rock medium at a constant temperature as:

$$c_r(p_l) := \frac{1}{\phi} \frac{\partial \phi}{\partial p_l} \quad (3-9)$$

and is assumed to be constant over the range of pressures of interest and equal c_r . Now combining equation (3-3),(3-4),(3-5) and the compressibility definition we obtain:

For water

$$-\nabla \cdot \left(\alpha \rho_w \frac{k_{rw}}{\mu_w} k \left[\nabla p_o - \frac{\partial p_c}{\partial s_w} \nabla s_w - \rho_w g \nabla d \right] \right) + \alpha \rho_w \phi \left[s_w (c_w + c_r) \frac{\partial p_o}{\partial t} + \frac{\partial s_w}{\partial t} \right] - \alpha \rho_w q_w = 0, \quad (3-10)$$

For oil

$$-\nabla \cdot \left(\alpha \rho_o \frac{k_{ro}}{\mu_o} k (\nabla p_o - \rho_o g \nabla d) \right) + \alpha \rho_o \phi \left[(1 - s_w) (c_o + c_r) \frac{\partial p_o}{\partial t} + \frac{\partial s_w}{\partial t} \right] - \alpha \rho_o q_o = 0. \quad (3-11)$$

Given that the reservoir simulator used for the numerical analysis does not support any type of compressibility, in this case the above equations become:

For water

$$-\nabla \cdot \left(\alpha \rho_w \frac{k_{rw}}{\mu_w} k \left[\nabla p_o - \frac{\partial p_c}{\partial s_w} \nabla s_w - \rho_w g \nabla d \right] \right) + \alpha \rho_w \phi \frac{\partial s_w}{\partial t} - \alpha \rho_w q_w = 0, \quad (3-12)$$

For oil

$$-\nabla \cdot \left(\alpha \rho_o \frac{k_{ro}}{\mu_o} k (\nabla p_o - \rho_o g \nabla d) \right) - \alpha \rho_o \phi \frac{\partial s_w}{\partial t} - \alpha \rho_o q_o = 0. \quad (3-13)$$

Which is a system of coupled nonlinear equations.

Relative permeabilities are required when several phases in the porous medium are modelled. In the case when more than one phase is flowing simultaneously through a porous medium, it is expected that the flow of each phase to be slowed down by the resistance of the pore structure and additional by the presence of other phases. The resistance of the pore structure is expressed by the absolute permeability, denoted as K , while the relative permeability of a phase l , k_{rl} , accounts for the permeability

reduction in the presence of other phases. The relative permeability is a dimensionless quantity that ranges from zero to one, whereas the absolute permeability is a rock property and measured in m^2 or in Darcy units, $1\text{Darcy} \approx 9.87 \cdot 10^{-13}m^2$. The effective permeability of a phase is the product of relative permeability and the absolute permeability. The relative permeability of the phase depends on the phase saturation. If the relative permeability function is plotted, its shape is mainly influenced by the wettability of the rock. The phase that preferentially sticks to the rock is called the wetting phase, and the phase that preferentially flows in the center of the pores is the non wetting phase. It is usually assumed that when dealing with a three phase problem (i.e water, oil and gas) then water is the wetting phase, gas is the non wetting phase and oil is the intermediate phase. The relative permeabilities are considered as functions of their own saturations and they are derived from experiments in a laboratory. By definition the relative permeability is zero when the saturation of the phase equals its residual saturation which is defined as the largest saturation for which the phase does not flow. The sum of the relative permeabilities is usually less than one. The Corey model of the relative permeability, defines the relative permeabilities in the following way:

$$k_{rw} = k_{rw}^0 s_w^{n_w}, \quad (3-14)$$

$$k_{ro} = k_{ro}^0 (1 - s_w^{n_o}), \quad (3-15)$$

where

$$s = \frac{s_w - s_{wc}}{1 - s_{or} - s_{wc}} \quad (3-16)$$

and k_{ro}^0 and k_{rw}^0 are the end point relative permeabilities for oil and water respectively, n_o and n_w are Corey exponents, s_{or} is the residual oil saturation and s_{wc} is the connate water saturation. The relative permeability curves for the fluid properties that were used in the numerical simulations are presented in the next figure:

Previously, it was mentioned that the relative permeabilities are a function of their own saturation, this being said Figure (6-2) tells a straight forward story of what is expected to happen in the reservoir. Which is, that as water is injected, the saturation of water is expected to increase and the saturation of oil to decrease. This is a direct result of the fact that water is injected and just oil is extracted initially.

Capillary pressure is defined as the difference in pressure across the interface between two immiscible phases, in our case the wetting phase and the non wetting phase, which leads us to the following expression:

$$p_c = p_{non-wettingphase} - p_{wettingphase}. \quad (3-17)$$

There are several types of possible interfaces and for every type different chemicals act as the wetting and the non-wetting phase. In the case of the oil-water interface,

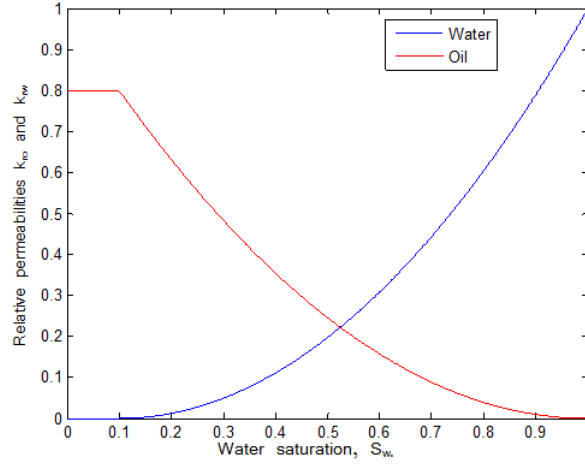


Figure 3-2: Corey type relative permeabilities.

water is typically the wetting phase, while at the oil-gas interface oil is typically the wetting phase. The reservoir simulator used for the numerical experiments, supports the option of accounting for capillary pressure, but the numerical results present in Chapter 6 do not account for capillary pressure.

The well model gives a relationship between grid block pressure and pressure in the well, also known as bottom hole pressure. This special modelling is required even though the average pressure and phase saturation in a unit volume grid block can be described by equations (3-12) and (3-13) this is not enough to represent the flow that takes place in the grid blocks where the wells are located. The well flow equations used in reservoir simulation are a set of algebraic equations:

$$q_t = J_p(p_{bh} - p_{gb}). \quad (3-18)$$

where q_t represents the total flow rate, p_{bh} is the bottom hole pressure, p_{gb} represents the grid block pressure, and J_p is the well index, which is defined as a function of saturation permeability, grid block geometry and reflects the effect of near-well flow. An array of constraints can be set up to describe how the well operates. The minimum and maximum bottom hole pressure, or maximum total flow rate can be fixed. In the case the bottom hole pressure constraint is active, using equation (3-18) the total flow rate q_t can be computed. In case of operating a well at fixed total rate, the bottom hole pressure can be calculated as:

$$p_{bh} = p_{gb} + J_p^{-1} q. \quad (3-19)$$

where q is the fixed total rate. For the numerical experiments a fixed total rate constrain is used.

3-4 State space representation of the reservoir flow model

After the spatial discretization on a chosen grid, the system of partial differential equation becomes a system of ordinary differential equations, which can be expressed as:

$$\begin{aligned} E(x(t))\dot{x}(t) &= A(x(t))x(t) + B(x(t))u(t), \\ x(0) &= x_{init}, \end{aligned} \quad (3-20)$$

where E is the accumulation matrix obtained after discretization of the accumulation term, $x(t)$ is a vector containing all unknown variables at time t , called the state vector, it is usually comprised of pressure and saturation vectors, $x^T = [p^T s^T]$, matrix A is the transmissibility matrix which is obtained after discretization of the transmissibility term, and u is the vector of well controls. $\dot{x}(t)$ represents the derivative of the state vector with respect to time, x_{init} represents the initial condition for the state vector. The initial value of the state vector is not known and is typically considered to be uncertain, but its uncertainty is considered to be smaller compared to the uncertainty in the parameters. For the purpose of this report the initial value of the state vector is considered to be perfect, something which is impossible in the case of a real life application. The matrices E, A, B are sparse matrices which are functions of state x and the differential equations that follow from them are non-linear. Matrix A depends on the state vector through the relative permeability term which as previously mentioned is a function of saturation, which in turn is part of the state vector. The matrix B , which is related to the sink/source term, is dependent on the state vector because in the well model J_p , the well index is also a function of saturation. In the case of realistic reservoir models the nonlinearities are larger because of more complex modelling of density, porosity, viscosity, or even permeability.

3-4-1 The discrete-time state space representation of the reservoir flow model

A commonly used time discretization in the reservoir simulation models is the fully implicit scheme, all dependent variables and inputs are evaluated at the next time step. Using Euler implicit time stepping, equation (3-20) becomes:

$$\begin{aligned} E(x(t_i)) - \delta A(x(t_i))x(t_i) &= E(x(t_i))x(t_{i-1}) + \delta B(x(t_i))u(t_i) \\ x(0) &= x_{init}, \end{aligned} \quad (3-21)$$

where t_{i-1} denotes the current time step, t_i denotes the next time step and $i \in 1, \dots, N$, where the whole simulation time length is t_N . This can be expressed in the following way:

$$\begin{aligned} g_i(x(t_i), x(t_{i-1}), u(t_i)) &= 0, \\ x(0) &= x_{init}. \end{aligned} \quad (3-22)$$

An iterative approach is needed in order to solve this system. It is common to solve the non-linear system of flow equations by using the Newton-Raphson method. At

each simulation step a few Newton-Raphson iterations are required:

$$x^k(t_i) = x^{(k-1)}(t_i) - \left[\frac{\partial g_i}{\partial x(t_i)} \right]^{-1} g_i(x^k(t_i), x(t_{i-1}), u(t_i)), \quad (3-23)$$

$$x^0(t_i) = x(t_{i-1}).$$

where the Jacobian $\frac{\partial g_i}{\partial x(t_i)}$ is evaluated at $x^{k+1}(t_i)$, and k stands for a Newton-Raphson iteration number. When the specified stopping criteria are satisfied an iteration terminates, this means that the Newton-Raphson iteration has converged and the solution of the system at time t_i is $x^k(t_i)$. In the case the Newton-Raphson iteration fails to converge, then the time step size is reduced and the procedure is iterated again. In practice given the fact that computing the inverse of the Jacobian is computationally intensive, it is never formed, instead at every iteration a linear system of equations is solved. The fully implicit method is unconditionally stable and therefore is commonly implemented. However, taking a time step which is too large, may lead to many Newton-Raphson iteration or failure of convergence, and larger discretization errors. The reservoir system can be described by the implicit system of equation,

$$\begin{aligned} g_i(x(t_i), x(t_i), u(t_i), \theta) &= 0, \\ j_i(y(t_i), x(t_i), u(t_i), \theta) &= 0, \\ x(t_0) &= x_{init}. \end{aligned} \quad (3-24)$$

or the explicit system of equations,

$$\begin{aligned} x(t_i) &= f_i(x(t_i), u(t_i), \theta), \\ y(t_i) &= h_i(x(t_i), u(t_i), \theta), \\ x(t_0) &= x_{init}. \end{aligned} \quad (3-25)$$

where the functional f_i and g_i describe how the state vector $x(t_i)$ changes from time $t_i - 1$ to time t_i , θ is a vector of uncertain parameters, in the case of production optimization it can be a vector of control variables, $u(t_i)$ is a vector of well controls, x_{init} is a given initial state of the system and $y(t_i)$ represent the outputs of the system at time t_i , given by the observation operator h_i or j_i .

3-5 Adjoint Model for the Reservoir Model

The flow models are used to solve production optimization or parameter estimation problems. Both these problems can be seen as optimization of a scalar objective function.

$$J = J(x(t_1), \dots, x(t_N), \theta), \text{ or} \quad (3-26)$$

$$J = J(x(t_1), \dots, x(t_N), u), \quad (3-27)$$

Equation 3-26 represents the objective function for a parameter estimation problem and 3-27 represents the objective function for a production optimization problem.

In both cases N corresponds to the total number of time steps and t_N represents the simulation time. J is typically a function of production data, therefore depends on the state vector $x(t)$. Computing the gradient of the objective function with respect to control variables, requires the availability of the system's sensitivity, this is commonly obtained using the adjoint method. In the adjoint approach, first, a modified function $\bar{J}$ is obtained by adjoining the system equations to J , and is given by:

$$\bar{J} = J(x_1, \dots, x_N, u) + \sum_{n=1}^N \lambda_n^T g_n(x_{n,n-1}, u) \quad (3-28)$$

where λ_n represents the adjoint variables at time t_n and (for the moment) is an arbitrary vector. The total variation of $\bar{J}$ which is sum of the variation with respect to dx_n , for $n = 1, \dots, N$ and du , is

$$d\bar{J} = dJ + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_n} dx_n \right) + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_{n-1}} dx_{n-1} \right) + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \quad (3-29)$$

Changing the index in terms which involve $x(t_{n-1})$ results in:

$$d\bar{J} = dJ + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_n} dx_n \right) + \sum_{n=0}^{N-1} \left(\lambda_{n+1}^T \frac{\partial g_n}{\partial x_n} dx_n \right) + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \quad (3-30)$$

Rearranging the limits of the sums involving $x(t_n)$ results in:

$$\begin{aligned} d\bar{J} = dJ + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_n} dx_n + \lambda_{n+1}^T \frac{\partial g_{n+1}}{\partial x_n} dx_n \right) + \\ \lambda_0^T \frac{\partial g_1}{\partial x_0} dx_0 - \lambda_{N+1}^T \frac{\partial g_{N+1}}{\partial x_N} dx_N + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \end{aligned} \quad (3-31)$$

Since the initial conditions are assumed to be fixed for our model, the term $\lambda_0^T \frac{\partial g_1}{\partial x_0} dx_0$ disappears. The boundary term $\lambda_{N+1}^T \frac{\partial g_{N+1}}{\partial x_N} dx_N$ is required to be zero for any differential dx_N , which in turn implies that $\lambda_{N+1} = 0$. The first term in equation (3-28), representing the total variation of J , is the sum of variations with respect to all state variables x_n and controls u and it is:

$$dJ = \sum_{n=1}^N \left(\frac{\partial J}{\partial x_n} dx_n \right) + \frac{\partial J}{\partial u} du \quad (3-32)$$

It becomes

$$\begin{aligned} d\bar{J} = \sum_{n=1}^N \left(\frac{\partial J}{\partial x_n} dx_n \right) + \frac{\partial J}{\partial u} du + \\ \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_n} dx_n + \lambda_{n+1}^T \frac{\partial g_{n+1}}{\partial x_n} dx_n \right) + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \end{aligned} \quad (3-33)$$

Rearranging the terms results in:

$$d\bar{J} = \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial x_n} + \lambda_{n+1}^T \frac{\partial g_{n+1}}{\partial x_n} + \frac{\partial J}{\partial x_n} \right) dx_n \quad (3-34)$$

$$+ \frac{\partial J}{\partial u} du + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \quad (3-35)$$

The adjoint model is obtained by setting all coefficients of dx_n to zero, i.e.

$$\lambda_n^T \frac{\partial g_n}{\partial x_n} + \lambda_{n+1}^T \frac{\partial g_{n+1}}{\partial x_n} + \frac{\partial J}{\partial x_n} = 0 \quad (3-36)$$

Taking the transpose of the above equations gives the system of adjoint equations

$$\left(\frac{\partial g_n}{\partial x_n} \right)^T \lambda_n + \left(\frac{\partial g_{n+1}}{\partial x_n} \right)^T \lambda_{n+1} + \left(\frac{\partial J}{\partial x_n} \right)^T = 0 \quad (3-37)$$

for $n = 1, \dots, N$. This system is solved backward in time because of the given ending condition, $\lambda_{N+1} = 0$, which can also be regarded as the initial condition for the adjoint system. If λ_n satisfies equation (3-37) then the total variation of $\bar{J}$ are given by:

$$d\bar{J} = \frac{\partial J}{\partial u} du + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} du \right) \quad (3-38)$$

it follows that the total derivatives of the objective function with respect to u are given by:

$$\frac{d\bar{J}}{du} = \frac{\partial J}{\partial u} + \sum_{n=1}^N \left(\lambda_n^T \frac{\partial g_n}{\partial u} \right) \quad (3-39)$$

It is important to observe that the reservoir equations are non-linear, on the other hand the adjoint equations are linear. The Newton-Raphson method requires few iterations to converge at each time step of the reservoir flow model, whereas, the adjoint model uses the sensitivity of the system at the converged solution of the Newton-Raphson method and requires only one system of equations at each time step to be solved.

3-5-1 Difficulties implementing an adjoint model

In a real life application, in order to capture and represent the complex dynamics of the physical system, the reservoir simulator used needs to be able to model a high number of details and becomes a very advanced and complex system. A reservoir simulator that solves a non-linear system of equations, offers an advanced modelling solution and consists of numerous lines of code which are implemented by a team of people, over a long period of time. In the case of using such a simulator the implementation of the adjoint method becomes a difficult task, because coding the partial derivatives of the system requires detailed knowledge of the simulation code. Even if the sensitivity matrices with respect to the current state vectors are available, the sensitivity matrices with respect to parameters can be very complicated

to derive. On top of that, the sensitivity matrices with respect to the previous state vectors are not implemented, the same is true for the sensitivities of the objective function. At present, if all partial derivatives of the system are accessible, then the adjoint code becomes relatively straightforward. Until recently computer storage capacity was also an issue of the adjoint implementation. This was due to the fact, that in order to solve the adjoint equations it was necessary to save the state vectors of the forward simulations. One way to work around this was to store only a limited number of state vectors and by recomputing the remaining ones. Now, all the simulated state vectors can be stored to disk and the storage problem disappears.

This is only possible in the case access to the simulator source code is available, which is not a very easy thing to obtain for commercial reservoir simulators. For the purpose of this thesis MRST-Sintef, reservoir simulator was used for all the numerical experiments. MRST represents an open-source Matlab toolbox that can be used as an efficient test platform for researching new optimization methods in reservoir simulations. The aim of MRST is to provide reproducible research and simplify the development, verification, validation and comparison of new methods on general grids. MRST contains a set of data structures and routines for creating, manipulating and visualizing petrophysical data, as well as routines for computing single and multiphase incompressible flow. At the time the implementation was done the adjoint-based gradient for controls and other parameters could not be accessible and because of this, the study focuses on methods that do not require access to the source code and are derivative free.

Reduced Order modelling

modelling physical systems usually results in complex high-order dynamic models. The amount of information contained in a high-order dynamic model may hide simple, insightful behaviours that need observation and quantification. It is often desirable to replace (approximate) these models by simpler models with reduced order, which can better capture and explore the simple behaviours that are needed. In this process, it is important to design the reduced model as to capture the important dynamics of the original high-order model. This chapter presents the model reduction technique that was used for the numerical experiments: Proper Orthogonal Decomposition (POD), is a method based on the projection of high-order system equations onto a subspace derived from snapshots. What snapshots are exactly and how they can be constructed is also described in this chapter.

4-1 Motivation

Simulations or, more generally, computational science has become an important part of today's technological world, and it is now generally accepted as the third discipline, besides the classical disciplines of theory and (real) experiment. Physical (and other) experiments lead to theories that can be validated by performing additional experiments. Predictions based on the theory can be made by performing virtual experiments, as illustrated by Figure 4-1.

Computer simulations are now performed routinely for many physical, chemical and other processes, and virtual design environments have been set up for a variety of problem classes in order to ease the work of designers and engineers. In this way, new products can be designed faster, more reliably, and without having to make costly prototypes. Originally, Reduced Order modelling (ROM) or Model Order Reduction (MOR) was developed in the area of systems and control theory, which studies properties of dynamical systems in application for reducing their complexity, while preserving their input-output behavior as much as possible. In the oil and

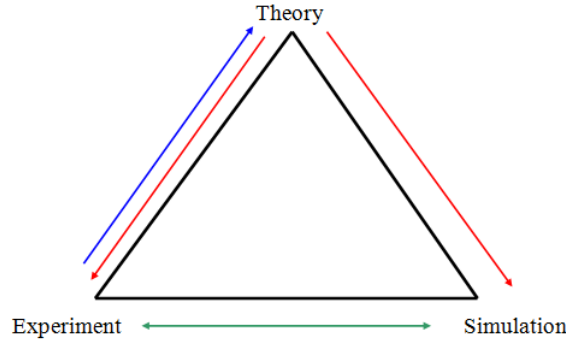


Figure 4-1: Simulation is the third big discipline.

gas industry ROM techniques provides a computationally efficient prototyping tool that replaces large scale models with approximate reduced order models, that are capable of capturing the essential dynamical behaviour and also preserve critical properties of the large models. A good model reduction technique will try to focus on the necessary information and ignore irrelevant information that comes from the high order model. One of the main reasons for using model reduction techniques is computational speed-up of simulation time. As a consequence, a simulation might finish in hours instead of days which is customary for some reservoir cases. Re-evaluation of methods which were prior considered too computationally expensive could be done or results could be improved by allowing for more simulations. An accurate and effective reduced-order model can be applied then for production optimization. In this thesis, the reservoir model is used for production optimization, but can also be used for history matching or prediction uncertainty quantification. Due to the fact that production optimization is the main focus, a very accurate representation of all state variables is not of main interest, but rather a good approximation of the production data. Therefore, a significant compression of the high-dimensional model is achieved. The conventional application of the ROM for the state-space reduction, reduces the computational time of the model simulation, but does not change the complexity of the model and as a consequence does not solve the implementation problem of the adjoint model. This thesis presents a non-standard application of model reduction where an approximate reduced order version of the tangent linear reservoir model is obtained. The reduced model is constructed by projecting the Jacobian of the model (which is approximated using finite differences), onto the reduced order space. Detailed explanations regarding the implementation process can be found in Chapter 5.

4-2 Projection-Based Reduced Order modelling Methods

The POD method is also known as the Karhunen-Loeve method, principal component analysis (PCA) or the method of empirical orthogonal functions. The origin of this method goes back to 1946 when it was introduced independently by Karhunen

and Loeve as a statistical tool to analyse random process data. The method was first called POD by Lumley (1967) when it was used for the study of turbulent flow. Lumley also traced the idea of POD back to independent investigations by Kosamby (1943), Pougachev (1953) and Obukhov (1954). It has been used in various disciplines including random variables Papoulis (1965), image processing Rosenfeld and Kak (1982), signal analysis, data compression and it has been successfully applied to increase the computational efficiency of the reservoir simulation by Heijn et al. (2004) Markovinovic et al (2002), Markovinovic and Jansen (2006) and Cardoso et al (2009). The use of POD in combination with adjoint based production optimization has been investigated by Van Doren et al (2005). The POD method possesses some desirable features for our application, such as: it allows non-linearities in the system, it is suitable for model reduction of spatio-temporal systems and it is feasible for large-scale systems.

The POD method is classified as a projection based method. The main idea behind projection based, reduced-order modelling is to project the high dimensional state space X onto the low dimensional subspace that contains the essential dynamics of the system. A projection is a linear transformation and can be represented by the use of projection matrix $\Pi \in R^n \times R^n$, which satisfies the condition $\Pi = \Pi\Pi^T$. A projection Π is defined as an orthogonal projection, if it is self-adjoint, i.e. $\Pi = \Pi^T$. A Galerkin projection Π is an orthonormal projection defined as, $\Pi = \Phi\Psi^T$, where $\Phi, \Psi \in R^{n \times l}$, $\Psi^T\Phi = I_l$, $\Psi = \Phi$, and the columns of Φ come from an orthogonal set. In the case that $\Phi \neq \Psi$, the projection matrix becomes a Petrov-Galerkin projection. The various projection based techniques differ in the choice of the matrices Ψ and Φ . Generally three main strategies, can be identified, for choosing Ψ and Φ :

- Heuristic schemes (Include choosing Ψ and Φ based on eigenvector analysis).
- Method based on rigorous system-theoretic analysis (Balanced Truncation).
- Parameter matching techniques (Krylov methods), for more details see Antoulas (2005) .

A projection based method consists of the following steps:

- Determining the l -dimensional subspace S_Φ and S_Ψ and their basis Ψ and Φ .
- Compressing the state information by projecting the state variables onto a subspace spanned by the basis Φ .
- Compressing the system equations by multiplying them by Ψ^T .

In order to exemplify the above steps, let us consider the Hilbert space $X \subset R^{n \times n}$, where n is the number of state variables needed to describe the system. Let the dynamic of the system be modeled by the following equations:

$$\begin{aligned} x(t_i) &= f_i[x(t_{i-1}), u], \\ y(t_i) &= h_i[x(t_i), u], \\ x(t_0) &= x_{init}, \end{aligned} \tag{4-1}$$

where $i = 1, \dots, N$ indicates the time steps, $x(t_i)$ is a solution of a functional f_i at time t_i , $x(t_0) = x_{init}$ is a given initial state of the system and $y(t_i)$ are the outputs of the system at time t_i , provided by the observation operator h_i . It is assumed that for all time instances $x(t_i)$ belongs to the Hilbert space X . For given matrices Φ and Ψ the system described by equations (4-1) is approximated by the reduced order system in the following form:

$$\begin{aligned} z(t_i) &= \Psi^T f_i[\Phi z(t_{i-1}), u], \\ y(t_i) &= h_i[\Phi z(t_i), u], \\ z(t_0) &= \Psi^T x(t_0), \end{aligned} \quad (4-2)$$

where $z(t_i)$ is the reduced-order state variable which operates in the subspace $S_\Phi \subset X$ of lower dimension l spanned by the basis Φ , and $z(t_0)$, represents the initial state of the reduced order system. The reduced order system equations (4-2) describe the dynamics of the l -dimensional reduced state vector z . Given that the reproduction of the dynamics involved in the original system is conditioned by the choice for matrices Φ and Ψ , this choice becomes the main obstacle for the ROM method.

4-2-1 Snapshots

In the POD method the projection subspace S_Φ is obtained by processing data from numerical simulations of the full order model described in (4-1). This data represents selected state vectors of a forward simulation of the full order model, called *snapshots*. The information present in the selected data should provide relevant insight about the dynamical behaviour of the system, and should be able to represent the behaviour of the system for different spacial and time instants. The general definition for snapshots, is the following:

$$x_j(t_i) = f_i[x(t_{i-1}), u] \quad (4-3)$$

where $t_i = t_1, \dots, t_N$ and u is a parameter vector and $x(t_{i-1})$ is a state vector determined by u . In order to use the reduced-order model for parameter estimation. The snapshots used for constructing the reduced-order model should be able to represent the dynamics of the system for modified parameter values. This results in the following snapshot.

$$x_j(t_i) = f_i[x^{pert}(t_{i-1}), u + u_{pert}] \quad (4-4)$$

where u_{pert} is a perturbation to the initial parameter values and $x^{pert}(t_{i-1})$ is a state vector determined by the perturbed parameters. These snapshots are the sampled solutions of the numerical model at particular time steps and at particular parameter values. Putting all snapshots as columns in a matrix the following matrix is constructed:

$$X = \{x_1(t_1), \dots, x_1(t_N), \dots, x_p(t_1), \dots, x_p(t_N)\} \quad (4-5)$$

where $X \in R^{n \times s}$, n represents the dimension of the state vector and $s = N \times p$, is the total number of snapshots, where p is the number of parameter perturbations. Generally it is true that $s \ll n$.

4-2-2 Projection matrices

As previously stated, the main issue of the POD method is expressing the essential information that is present in the collected data by means of a few basis vectors Φ . POD provides an optimal set of basis vectors, it guarantees the best reconstruction of the data set X in the mean square error sense among all linear bases of the same size. As it was stated, the aim of the POD is to find an orthogonal basis Φ , which contains l , n -dimensional vectors, such that the total mean squared error between the snapshots and their projections on the subspace spanned by the basis Φ is minimized for any l , i.e. the given value.

$$\frac{1}{s} \sum_{i=1}^s \|x_i - \Phi \Phi^T x_i\|_2^2, \quad (4-6)$$

is minimal, where $\|\cdot\|$ is an L_2 norm and $x_1 \in X$, $i = 1, \dots, s$ where $1 \leq s \ll n$ is assumed. This problem can be expressed as:

$$\min_{\Phi: \phi_i^T \phi_j = 0, \forall i \neq j, \|\phi_i\|_2^2 = 1} \frac{1}{s} \sum_{i=1}^s \|x_i - \Phi \Phi^T x_i\|_2^2. \quad (4-7)$$

The second norm in (4-7) can be written as:

$$\|x_i - \Phi \Phi^T x_i\|_2^2 = \|x_i\|_2^2 - \|\Phi \Phi^T x_i\|_2^2 \quad (4-8)$$

which holds for any orthogonal projection, the minimization problem (4-7) is equivalent to the following maximization problem.

$$\max_{\Phi: \phi_i^T \phi_j = 0, \forall i \neq j, \|\phi_i\|_2^2 = 1} \sum_{i=1}^s \|\Phi \Phi^T x_i\|_2^2 \quad (4-9)$$

where each element $\|\Phi \Phi^T x_i\|_2$ of the sum can be rewritten as:

$$\begin{aligned} \|\Phi \Phi^T x_i\|_2 &= \sqrt{(\Phi \Phi^T x_i)^T (\Phi \Phi^T x_i)} \\ &= \sqrt{x_i^T \Phi \Phi^T \Phi \Phi^T x_i} \\ &= \sqrt{x_i^T \Phi \Phi^T x_i} \\ &= \|\Phi^T x_i\|_2 \end{aligned} \quad (4-10)$$

Therefore, problem (4-7) is equivalent to the following maximization problem.

$$\max_{\Phi: \phi_i^T \phi_j = 0, \forall i \neq j, \|\phi_i\|_2^2 = 1} \sum_{i=1}^s \|\Phi^T x_i\|_2^2 \quad (4-11)$$

which is a constrained optimization problem that can be solved by considering the first-order necessary optimality conditions. First, this problem is solved for the case when $l = 1$, i.e. Φ consists of a single vector ϕ . Let L be the Lagrangian functional associated with equation (4-11)

$$L(\phi, \lambda) = \sum_{i=1}^s \|\phi^T x_i\|_2^2 + \lambda(1 - \|\phi\|_2^2). \quad (4-12)$$

A first-order necessary optimality condition is given by

$$\nabla L(\phi, \lambda) = 0, \quad (4-13)$$

the derivative of L with respect to ϕ is:

$$\begin{aligned} \frac{\partial L}{\partial \phi}(\phi, \lambda) &= \frac{\partial}{\partial \phi} \left(\sum_{i=1}^s (\phi^T x_i)^T (\phi^T x_i) + \lambda(1 - \phi^T \phi) \right) \\ &= \frac{\partial}{\partial \phi} \left(\sum_{i=1}^s (x_i^T \phi \phi^T x_i) + \lambda(1 - \phi^T \phi) \right) \\ &= 2 \sum_{i=1}^s \phi^T x_i x_i^T - 2\lambda \phi^T \\ &= 2(\phi^T X X^T - \lambda \phi^T) \end{aligned} \quad (4-14)$$

Now what we get is that:

$$2(\phi^T X X^T - \lambda \phi^T) = 0 \quad (4-15)$$

which provides the eigenvalue problem:

$$(\phi^T X X^T - \lambda \phi^T)^T = 0 \iff X X^T \phi = \lambda \phi \quad (4-16)$$

This means that ϕ is the eigenvector corresponding to the largest eigenvalue from problem (4-16), and the maximum reached value is equal to λ . It can be proven that for any $l \in 1, \dots, s$ the solution of problem (4-7) is given by the first l largest eigenvectors of matrix $X X^T$. The first l largest eigenvectors correspond to the first l largest eigenvalues of the correlation matrix:

$$R := \frac{1}{s} X X^T. \quad (4-17)$$

Instead of the correlation matrix R , the covariance matrix can be used:

$$\bar{R} := \frac{1}{s-1} \sum_{i=1}^s (x_i - \bar{x})(x_i - \bar{x})^T \quad (4-18)$$

where $\bar{x}$, represents a mean of the snapshots, $\bar{x} = \frac{1}{s} \sum_{i=1}^s x_i$. The decision on how many eigenvectors should Φ contain, is based on the cumulative relative importance, defined as:

$$\sum_{i=1}^l \frac{\lambda_i}{\sum_{j=i}^s \lambda_j} > \alpha^e, \quad (4-19)$$

where l , is chosen as the smallest number such that,

$$\sum_{j=1}^l \alpha_j \geq \alpha^e, \quad \alpha_i = \frac{\lambda_i}{\sum_{j=1}^l \lambda_j} \quad (4-20)$$

Summing everything, the orthonormal projection matrix Φ is created by taking vectors ϕ_j as its columns for $j \in 1, \dots, l$. The accuracy of the reduced order model

can be increased by also increasing the number of vectors in the matrix Φ . The highest accuracy is obtained for $\alpha^e = 100\%$. On the other hand more patterns means more runs of the full order numerical model and a low computational efficiency. Therefore, there exists a trade-off between efficiency and accuracy.

Numerical issues

$XX^T \in R^{n \times n}$ is a symmetric matrix satisfying:

$$\phi^T(XX^T)\phi = (X^T\phi)^T(X^T\phi) \leq 0 \quad \forall \phi \in R^n \quad (4-21)$$

which means that XX^T is positive semi-definite, and therefore has non-negative eigenvalues, $\lambda_1 \leq \dots \leq \lambda_s \leq 0$ and corresponding s eigenvectors. For large system solving eigenvalue problem given by equation (4-20) is computationally demanding. Another way to calculate the POD patterns is to perform a singular value decomposition (SVD), on the snapshot matrix X . It circumvents calculating the product XX^T or X^TX which can be dense matrices. The matrix X is decomposed as:

$$X = U\Sigma V^T \quad (4-22)$$

where $U \in R^{n \times n}$, $V \in R^{s \times s}$, are orthogonal, meaning that, $U^TU = I_n$, $V^TV = I_s$ and $\sigma \in R^{n \times s}$ is a pseudo diagonal matrix with elements ordered in the decreasing manner. Looking at the correlation matrix XX^T , we see that:

$$XX^T = U\sigma V^T(U\Sigma V^T)^T = U\sigma V^TV\Sigma^TU^T = U\Sigma\Sigma^TU^T \quad (4-23)$$

It follows that the POD patterns Φ can be computed as left singular vectors U , whereas, $\lambda_i = \sigma_i^2$. They can also be expressed using right singular vectors, V , because for X^TX it holds:

$$X^TX = (U\Sigma V^T)^TU\Sigma V^T = V\Sigma^TU^TU\Sigma V^T = V\Sigma^T\Sigma V^T. \quad (4-24)$$

4-2-3 Reduced order model

The projection $\Pi = \Phi\Psi^T$ is defined as a Galerkin projection with $\Psi = \Phi$, where Φ consists of l eigenvectors, ϕ_i , corresponding to the first l highest eigenvalues. The state of the system can then be expressed in the new space as linear combination of the POD patterns, i.e. $x(t_i) = \Phi z(t_i)$ or $x(t_i) = \bar{x} + \Phi z(t_i)$, if the mean value was subtracted from the snapshots. Then the reduced order system is described by:

$$\begin{aligned} z(t_i) &= \Phi^T f_i(\Phi z(t_{i-1}), u) \\ y(t_i) &= h_i(\Phi z(t_i), u) \\ z(t_0) &= \Phi^T x(t_0), \end{aligned} \quad (4-25)$$

The reduced-order system captures the high-order dynamics that were present in the snapshots.

4-2-4 POD method comments

Using the POD method the high dimensional state space X is projected onto the low dimensional subspace that contains the essential dynamics of the system. Besides numerous advantages, the method also exhibits drawbacks.

- Optimality of a particular basis does not imply optimality of the reduced order dynamical model. The method is highly dependent on the given data. Therefore, it is essential to choose snapshots that represent the dynamics of the model in the range of the parameters or control variables that we are interested in. However, there is no guarantee that the reduced order model will follow the same dynamics as the high order model.
- Error bounds cannot be calculated, using the POD method. It is system representation dependent. There is no good measure to estimate the uncertainty of the solution provided by the reduced order model.
- The sparse structure of the partial differential equations is lost during the process of projection onto the reduced-order subspace. Leading to the conclusion that the dense structure of the reduced-order model does not imply extraordinary reduction in the computational time.
- In case of highly non-linear systems, the system matrices, $f_i(\Phi z(t_{i-1}), \theta)$ need to be recomputed at every time step and projected on the low order subspace, which is computationally expensive. Therefore, the POD method is more effective for linear cases, where system matrices have to be projected only once in order to capture the main dynamics of the system.

Model-Reduced Gradient Based Production Optimization

Gradient based production optimization algorithms can be used to update production strategies for a reservoir model using production data. However, they require the implementation of an adjoint model and sensitivity matrices of the reservoir model with respect to the controls. This usually implies a considerable programming effort. This thesis presents a new proposed approach to gradient-based production optimization done using model reduction. The original non-linear high order forward model is replaced by a linear reduced order forward model and the adjoint model of the tangent linear approximation of the full order forward model is replaced by the adjoint of a linear reduced order forward model. The construction of the reduced order model is done using the POD method. The reduced order model is not obtained by using a conventional projection based ROM technique, that projects the non-linear system equations on the reduced subspace. This requires access to the Jacobian with respect to the states and controls of the high order tangent model. For the purpose of this thesis it was assumed that access to the Jacobian was not possible. The approach studied obtains a reduced-order approximation of the tangent linear model by projecting approximate derivatives of the full order model. Then due to the linear characteristic of the reduced-order model, the corresponding adjoint model is easy to obtain. The gradient of the objective function is approximated and the maximization problem is solved, in the reduced space. The procedure can be iterated for the updated controls if necessary, to look for further improvements of the control vector. The POD method is adjoint free and can be implemented around any reservoir simulator. The method is compared with ensemble-based optimization, another stochastic gradient-based production optimization method, that is also adjoint free and can be used with any reservoir simulator without the need for access to the source code.

5-1 Introduction

Production optimization problem

As presented in Chapter 2, during the production optimization phase of the reservoir management process, one tries to identify the best position of wells and a configuration of controls that would provide a maximized production from the reservoir. For the purpose of this thesis the Net Present Value is considered. Describing the process as an optimization problem, finding a control sequence that would maximize the NPV, is defined as the solution to this problem.

$$J(x(t), u(t)) = \sum_p (r_o^p(t)q_o^p(t) - r_w^p(t)q_w^p(t)) - \sum_i r_w^i(t)q_w^i(t), \quad (5-1)$$

At every time step t the objective function depends on the state vector $x(t_i)$ which is constrained by the system equations

$$x(t_i) = f_i(x(t_i), u(t_i)) \quad (5-2)$$

where $x(t_0)$ represents the initial saturation and pressure in each grid block of the reservoir, and $u(t_0) = u_{init}$, represents the initial control strategy, which generally is a constant control strategy where the water injection remains constant over the life time of the reservoir.

5-2 Reparametrization

The number of elements in the control vector is calculated as the number of wells times the number of control times at which the control values can change. This leads to the dimensionality problem with respect to the control vector, meaning, that if the dimension of the case study grows (more wells and more control times), then also the dimension of the control vector grows. This is not a big computational burden for small theoretical studies. Because an approximation of the sensitivity of the full-order reservoir model with respect to the controls is required in order to construct the reduced model. This amounts to running the full order forward model once for every element in the control vector and to a high computational cost of building the reduced model. Also because the reduced order model, might be constructed several times during the optimization process what was observed is that a significant computational advantage is not achieved for complex reservoirs. For the purpose of this thesis, this problem was avoided by reparametrizing the control vector u in order to reduce the number of control variables. The method proposed and used for the numerical simulations is, again, the POD method. The control vector is approximated by:

$$u = u_{init} + \Phi_\eta \eta \iff \Delta u = \Phi_\eta \eta \quad (5-3)$$

where the control patterns creating matrix Φ_η are the first p_{red} dominant eigenvectors of a low-rank approximation of the covariance matrix of the control space, represented by an ensemble of prior controls and u_{init} is the mean of the ensemble. Alternatively, other reparametrization methods may be chosen, such as, the

discrete cosine transform Jafarpour and McLaughlin (2009) or a discrete wavelet transform Sahni and Horne (2005), as long as they lead to a significant reduction in the number of controls. Description of the two mentioned methods can be found in the literature review part of the thesis.

5-3 Gradient-Based Model-Reduced Production Optimization

Classical gradient-based production optimization formulated for a general model represents a non-linear constrained optimization problem, that is very difficult to solve. It can be simplified by considering that the objective function can be made quadratic by assuming that the, model, f_i and, observation, h_i operators can be linearized. The theoretical advantage of using a quadratic objective function is that it has a unique solution, and it can be found, solving only linear equations. Therefore, the tangent linear model of the original model is written using the first-order Taylor formula in the vicinity of the background control vector u^b . This results in the following equation:

$$\Delta \bar{x}(t_i) = \frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} \Delta \bar{x}(t_{i-1}) + \sum_{j=1}^p \frac{\partial f_i[x(t_{i-1}), u]}{\partial u_j} \Delta u_j \quad (5-4)$$

where p is the size of the control space and $\Delta \bar{x}$ represents the change in the state vector with respect to the change in controls. The partial derivatives in (5-4) are evaluated at, $(x^b(t_{i-1}), u^b)$, where $x^b(t_i) = f_i[x^b(t_{i-1}), u^b]$. Equation (5-4) can be rewritten as:

$$\begin{bmatrix} \Delta \bar{x}(t_i) \\ \Delta u \end{bmatrix} = \begin{bmatrix} \frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} & \frac{\partial f_i[x(t_{i-1}), u]}{\partial u_j} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta \bar{x}(t_{i-1}) \\ \Delta u \end{bmatrix} \quad (5-5)$$

which is a linear model in terms of variation. The maximization process is then performed by the following iterative steps:

- Linearize the model around the background control value u_k^b , where k denotes the iteration number and $u_1^b = u_{init}$,
- Find Δu_k such that it maximized the quadratic objective function.
- Update $u_{k+1}^b = u_k^b + \Delta u_k$

The production optimization procedure becomes a two loop scheme, containing a inner loop, which finds the maximum of the quadratic objective function and an outer loop, where the original model is used to redefine the model trajectory and to calculate the original objective function. It iterates until some predefined criteria is met. The computational speed-up sought after by the model reduction technique, is obtained by replacing the tangent linear model with a simpler low resolution linear model, constructed using the information from the perturbations of the original model. Because the reduced-order model is designed to capture

the dominant dynamics of the original system, it is expected that it can be used successfully for a fast maximization of the objective function in the inner loop. It is worth mentioning that there is no guarantee that the reduced-order model will always lead to the same response with respect to control variations as the full-order model.

5-3-1 Initialization

The initial control vector is chosen as a general constant injection rate strategy, meaning, that the injection rate in the injection well is kept constant during the life cycle of the reservoir. Next the high order reservoir simulation is run for the initial controls and the initial objective function is calculated.

5-3-2 Collection of the snapshots and pattern selection

As previously stated, the POD method can be used to obtain a low order version of the tangent linear approximation of the model. This is different than the conventional application of the POD method for state-space reduction. In the conventional approach, a data matrix is built from selected state vectors of a forward simulation of the full-order numerical model, called snapshots. Spatial basis functions are obtained by computing a low rank approximation of the covariance matrix of the data and by selecting the leading eigenvectors. Next the tangent linear model equations are projected on the low-order subspace, spanned by the basis vectors, which are still able to reproduce the dynamical behaviour of the original model Cardos et al (2009) or Van Doren et al (2004). If the original model is non-linear then the matrix coefficients of the tangent linear high-order model need to be recomputed every time step before applying the reduction, which significantly reduces the computational advantage of the POD method, as compared to the linear case. Furthermore, the conventional POD method requires access to the Jacobians with respect to states. The approach present in the numerical examples, aims at obtaining the reduced-order approximation of the tangent linear model directly by computing approximate derivatives of the reduced-order model. For the purpose of our application, the reduced-order model is used for production optimization therefore the snapshots need to be able to represent the dynamics of the system for modified control values. The snapshots are created in the following way:

$$x_j(t_i) = f_i[x^b(t_i, u^b + \phi_\eta^j \eta_j)] - f_i[x^b(t_{i-1}), u^b] \quad (5-6)$$

where ϕ_η^j is the j th pattern in matrix Φ_η and η_j is the size of the perturbation. This snapshots are ordered as columns in matrix:

$$X = \{x_1(t_1), \dots, x_1(t_N), \dots, x_{p_{red}}(t_1), \dots, x_{p_{red}}(t_N)\} \in R^{n \times s} \quad (5-7)$$

where s is the total number of snapshots, n represents the dimension of the state vector and p_{red} is the dimension of the control space after reparametrization. The given space of snapshots spans a subspace on which the reduced-order model is

built. Since typically $s \ll n$, instead of solving the eigenvalue problem for matrix $XX^T \in R^{n \times n}$ the reduced eigenvalue for matrix $X^T X \in R^{s \times s}$ is solved.

$$(X^T X)w_i = \lambda_i w_i, \quad i \in 1, \dots, s \quad (5-8)$$

The eigenvalues for both problems are the same and the following relationship holds

between the eigenvectors: $\phi_i = \frac{1}{\sqrt{\lambda_i}} X w_i$ for $i \in 1, \dots, s$. The eigenvectors corresponding to the largest eigenvalues of matrix XX^T represent the dominant directions present in the snapshots, called patterns. Each pattern has a relative importance associated with it, α_i , which is defined as the percentage of the eigenvalues in the total sum of eigenvalues, $\alpha_i = \frac{\lambda_i}{\sum_{j=1}^s \lambda_j}$. By choosing n_{red} as the number of patterns with the largest relative importance, an orthonormal projection matrix Φ is created, where $n_{red} \leq s$. n_{red} is chosen as the smallest number such that $\sum_{j=1}^{n_{red}} \alpha_j \geq \alpha^e$, where α^e is some percentage of the total relative importance.

5-3-3 Building the reduced-order linear model

A model can be reduced by reconstructing the incremental state $\Delta \bar{x}(t_i)$, according to

$$\Delta \bar{x}(t_i) = \Phi z(t_i) \quad (5-9)$$

where Φ is the projection matrix derived in the previous subsection and $z \in R^{n_{red}}$ is the reduced state vector. Combining equations (5-4), (5-3) and (5-9) results in:

$$z(t_i) = N_i z(t_{i-1}) + N_i^u \eta \quad (5-10)$$

where

$$\begin{aligned} N_i &= \Psi^T \frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} \Phi \\ N_i^u &= \Psi^T \frac{\partial f_i[x(t_{i-1}), u]}{\partial u} \Phi_\eta \end{aligned} \quad (5-11)$$

and where z represents the response of the original model to a perturbation of the controls, such that we can write:

$$\begin{bmatrix} z(t_i) \\ \eta \end{bmatrix} \begin{bmatrix} N_i & N_i^u \\ 0 & I \end{bmatrix} \begin{bmatrix} z(t_{i-1}) \\ \eta \end{bmatrix} \quad (5-12)$$

The partial derivatives in equation (5-12) represent the sensitivity of the model with respect to the states and to the controls, but can not be easily computed explicitly and should be approximated by:

$$\left[\frac{f_i[x(t_{i-1}), u]}{\partial u} \right]_j \approx \frac{f_i[x(t_{i-1}), u + \epsilon_u i_j^u] - f_i[x(t_{i-1}), u]}{\epsilon_u} \quad (5-13)$$

and

$$\left[\frac{f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} \right]_j \approx \frac{f_i[x(t_{i-1}) + \epsilon i_j, u] - f_i[x(t_{i-1}), u]}{\epsilon} \quad (5-14)$$

where ϵ and ϵ_u are the intervals in which the partial differential equations are linearized and i_j and i_j^u are the j th column of the identity matrices, $I^{n \times n}$ and $I^{p \times p}$. These perturbations are too expensive for large-scale models. Looking at equations (5-12), here is no need to calculate those partial derivatives explicitly and instead of approximating terms $\frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})}$ the term $\frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} \Phi$ is approximated as:

$$\frac{\partial f_i[x(t_{i-1}), u]}{\partial x(t_{i-1})} \phi_j \approx \frac{f_i[x(t_{i-1}) + \epsilon \phi_j, u] - f_i[x(t_{i-1}), u]}{\epsilon}, \quad (5-15)$$

where ϕ_j represents the j th patterns from matrix Φ . Following the same scheme we approximate the term $\frac{\partial f_i[x(t_{i-1}), u]}{\partial u} \Phi_\eta$ as:

$$\frac{\partial f_i[x(t_{i-1}), u]}{\partial u} \phi_\eta^j \approx \frac{f_i[x(t_{i-1}), u + \epsilon_u \phi_\eta^j] - f_i[x(t_{i-1}), u]}{\epsilon_u}, \quad (5-16)$$

where ϕ_η^j is the j th column of Φ_η . It results in a reduced-order model which is linearized along patterns. The dimension of the reduced-order model depends on the number of patterns and on the number of variables p_{red} , that need to be estimated, which are both expected to be small. The matrices that are involved in the reduced-order model are known explicitly and the reduced-order model is linear. Therefore, the computational time for one reduced-order simulation can be neglected. However, the computational time required for constructing reduced-order model may be significant.

So far, the description of the procedure involved in building the reduced-order model, was presented. The formulation of the method was presented using an explicit form of the reservoir equations, which was motivated by the resulting simplification in the description of the method. An explicit method calculates the state of the system at the previous time step, while an implicit method finds a solution by solving an equation involving both the current state of the system and the previous one. Since the tangent linear model for both cases can be rewritten such that it calculates the approximate state at a current time step from the approximate state at the previous time step, and from the parameters, the choice of an explicit or an implicit method is irrelevant.

5-3-4 Improving parameters in the reduced space

The moment when the reduced-order approximate linear model is available, the adjoint model is easy to implement. A gradient-based optimization technique is available, where the gradients are obtained with the aid of an adjoint equation in terms of Lagrange multipliers $\lambda(t_i)$. The multipliers represent the objective function sensitivities to changes in the state variables and originate from adding the dynamic system as a constraint to the objective function. In our application, the controls are formed by the injection rates in the injection wells at the control times. The adjoint equation can be written in discrete time as:

$$\lambda_{t_i}^T \left(\frac{\partial g_{i+1}}{\partial x(t_i)} \right) = \left[-\lambda(t_{i+1})^T \frac{\partial g_i}{\partial x(t_i)} - \frac{\partial J(x(t_i), u(t_i))}{\partial x(t_i)} \right], \quad (5-17)$$

where

$$g_i = x(t_{i-1}) - f_i(x(t_{i-1}), u(t_i)) = 0 \quad (5-18)$$

is a compact representation of the system equation. In the implementation, instead of the full-order model, the reduced-order model is added as a constraint to the objective function J with the aid of a set of low-order Lagrange multipliers μ ,

$$\bar{J}_{red}(z(t_i), \Phi_u u(t_i)) = \sum_{i=0}^{N-1} [J(\Phi z(t_i), u(t_i)) + \mu(t_{i+1})^T g(\Phi z(t_{i+1}), \Phi z(t_i), u(t_i))] \quad (5-19)$$

where the average of the state vector $\bar{x}$, needs to be added to each product Φz . Taking the first variation of equation (5-19), and reworking the results, a reduced-order equation in terms of reduced-order Lagrange multipliers is obtained:

$$\underbrace{\mu(t_i)^T \Phi^T \frac{\partial g_{i-1}}{\partial x(t_i)} \Phi}_{p \times p} = \left[-\underbrace{\mu(t_{i+1})^T \Phi^T \frac{\partial g_i}{\partial x(t_i)} \Phi}_{p \times p} - \underbrace{\frac{\partial J(x(t_i), u(t_i))}{\partial x(t_i)} \Phi}_{p \times p} \right] \quad (5-20)$$

Starting from the final condition $\mu(t_N) = 0$ it can be integrated backward in time. Because the derivative in equation (5-20) consist of state dependent parameters they are approximated using finite differences, as previously shown. Then they are projected onto the low-order subspace. After calculating μ for every time step, we can calculate:

$$\frac{\partial L(t_i)}{\partial \eta(t_i)} = \frac{\partial J(z(t_i), \eta(t_i))}{\partial \eta(t_i)} + \mu(t_{i+1})^T \left\{ \Phi^T \frac{\partial g_i}{\partial \eta(t_i)} \right\} \quad (5-21)$$

. The updated reduced controls are computed using a steepest ascent method, $\eta_{k+1} = \eta_k + \epsilon \frac{\partial L(t_i)}{\partial \eta(t_i)}$, where ϵ is a weighting factor, that is adjusted during the maximization process.

5-3-5 Improving controls for the full-order model

The solution provided by the reduced-order model control optimization process, represents an optimum for the reduced-order linearized system, but not necessary for the original full order model. The value of the original objective function with respect to the new controls, $u_{k+1}^b = u_k^b + \Phi_\eta \eta_{opt}$, where η_{opt} represents the solution of the reduced order maximization process. The optimization process stops when either there is no improvement in the objective function from one iteration to the next, or a specified number of iteration is reached.

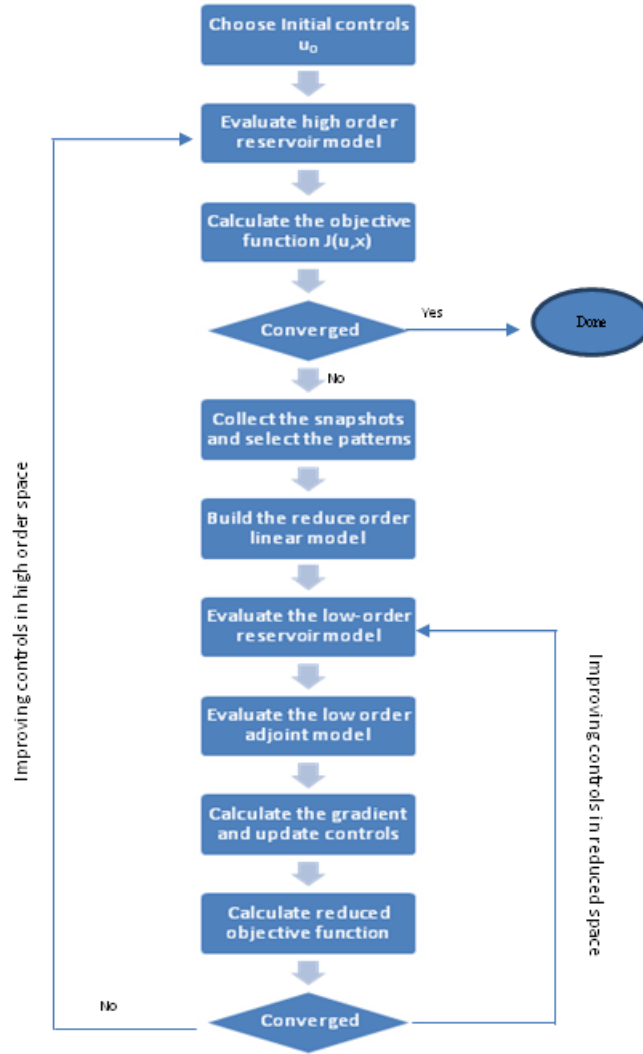


Figure 5-1: Model-reduced gradient-based production optimization

If the objective function does not obey the stopping criterion, then the process restarts around a new background control vector u_{k+1}^b . Since there is a relationship between patterns and model parametrization, whenever the background parameter is changed, a new background state and a new set of patterns needs to be identified. This lead to the construction of a new reduced order-model, which in turn lead to a new inner loop run, that performs a new optimization of the reduced-order control vector. The production optimization algorithm presented in this chapter can be summarized by the flow chart in Figure (5-1).

5-3-6 Computational complexity

The computational cost of the reduced-order production optimization approach can be split in two main parts: the cost of constructing the reduced order model, and the cost of solving the reduced system and the production optimization problem.

The Cost of constructing the reduced order model

- Generating "good" snapshots is an essential part of the POD method, which is a data driven method. The actual generation of the snapshots is done by generating an ensemble of controls around a initial control. For each member in the newly generated ensemble the full-order model is ran, and the values for pressure and saturation at each time step are saved. The saved values are then subtracted from the pressure and saturation values of a full-order model run with the background controls u^b . The computational cost of performing this part of the optimization process, expressed in full-order model runs is equal with the number of members in the generated ensemble.
- The cost of solving the reduced eigenvalue problem is equivalent to the cost of a Singular Value Decomposition of matrix X in case of POD method. Since the number of snapshots can be kept low, this cost is also low.
- Cost of approximating partial derivatives along patterns, represents the most computationally expensive part of the optimization procedure based on model-order reduction. The computational time expressed in full-order model runs is equal to the number of patterns generated. Let P be the total number of patterns, then $P = p_{red} + n_{red}$, where p_{red} , represents the number of control patterns, and n_{red} , is the number of state patterns.

The cost of solving the reduced optimization problem

- The cost of solving a system of n_{red} linear equations is equal to $O(n_{red}^2)$, which can be neglected in our case.
- The cost of the reduced optimization procedure, is limited to the number of times the new controls constructed for the purpose of improving the reduced objective function are verified to also improve the original objective function. This requires a full order model run for each original objective function calculation.

5-3-7 Additional issues

Pressure and saturation variables have different physical behaviour, a different order of magnitude and generally a different variability in their values. The POD procedure favours variables which values show grater variability, and this can have negative effects on the pattern identification. Following the information available in the literature, it was decided that snapshot collection for pressure and saturation should be done separately. Instead of solving one eigenvalue problem to obtain the POD matrix Φ , two separate eigenvalue problems were solved, one for pressure snapshots, and one for saturation snapshots. Later, because MRST, the reservoir simulator used for the numerical experiments, works with incompressible flow, it was observed that approximating the partial derivatives along the patterns for pressure

provided no information that would be useful for the reduced-order model. Given this circumstance, the decision to ignore the pressure patterns was taken, leaving us with a reduced model built only using saturation patterns.

5-4 Ensemble-Based Optimization

The ultimate goal of production optimization is the development of an improved operating plan, called field development plan (FDP), which has the potential to substantially increase the ultimate oil recovery. The NPV and the ultimate recovery are considered the objective functions in an optimization problem, and are the quantities that need to be maximized in order for the production optimization process to be successful. Because the NPV is an economic objective calculated based on evaluations of future production behaviour, a reservoir simulation model is required to provide this forecasted production values. Due to the fact the reservoir is not easily observable, the reservoir geological model is possible to be highly uncertain. In order to obtain a suitable production strategy for the reservoir of interest, production optimization needs to be combined with a parameter estimation method that reduces the uncertainty of the estimate of the reservoir geological properties in a closed loop optimization Chen et al. (2008). In the approach presented in this thesis it is assumed that the geological properties are known and no parameter estimation method is implemented.

5-4-1 Ensemble optimization

The ensemble optimization approach, is based on the work of Chen et al (2008). It uses the statistical information from an ensemble of random perturbations of control variables, so that the cross covariance between the objective function and the control variables can be estimated. The cross-covariance needs to be computed because it contains the information regarding the correlation between the control variables and the objective function. In laymen terms it explains that if a control variable is positively correlated with the objective function (this is denoted by a positive value in the covariance matrix). It is expected that an increase in that control variable will result in an increase in the objective function. For the numerical experiments present in Chapter 6, EnOpt has the same goal as MorOpt, which is to maximize the net present value of the reservoir. The net present value has the same expression as presented in Chapter 1. The vector of control variables u , also known as the control vector, contains all the well constraints at different control steps and is represented in the following way:

$$u = [u_1, u_2, u_3, \dots, u_{N_u}] \quad (5-22)$$

where N_u is the total number of control variables, and is computed as the product of the number of wells and the number of control steps. Let x be a typical state vector for a two phase flow problem, for each grid block of the underlying geological model x contains porosity, permeability, pressure and water saturation and all the

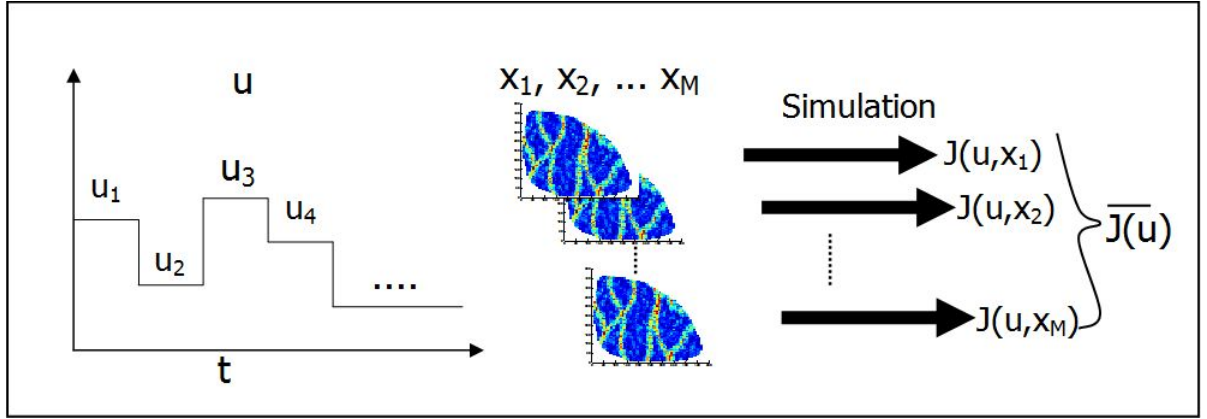


Figure 5-2: Expected Value of the Objective Function

available production data, water production rate and oil production rate.

$$x = [\phi, \ln k, P, S_w, WPR, OPR]^T \quad (5-23)$$

The objective function, J , is going to be a function that depends on the control vector (see 5-22) and on the state vector (see 5-23), $J = J(u, x)$, because the oil and water production rates both depend on the control vector and on the reservoir properties. In a practical application, because the reservoir is deep underground, the geological properties of the reservoir are subject only to indirect observations. This in turn generates uncertainty, which is assumed to be captured by a set $x = [x_1, \dots, x_M]$ representing the state vectors of M equiprobable models. The optimization goal becomes maximizing the expected value of the objective function over the uncertain description of the model parameters. The optimization problem becomes:

$$\begin{aligned} & \max_u \overline{J(u)} \\ \text{subject to } & u_{low} \leq u \leq u_{up} \end{aligned} \quad (5-24)$$

where $\overline{J(u)}$ is given by

$$\overline{J(u)} = \frac{1}{M} \sum_{i=1}^M J(u, x_i) \quad (5-25)$$

As previously stated, for the numerical experiments, the the uncertainty of the model parameters is not considered, the model is considered to be exact. This means that in the previously presented optimization problem the models are identical and the problem reduces to a deterministic optimization problem.

5-4-2 Gradient optimization

Before further introducing the stochastic gradient. It is useful to describe exactly how the classical gradient is used in an optimization process. The objective function is assumed to be "smooth" meaning is once or several times differentiable with respect to the controls. When considering problems which deal with "smooth" objective function, it is of interest to consider some calculus based aspects of analysis. The gradient of the gain function is defined as the N_c -dimensional column vector.

$$\nabla J(u) = \frac{\partial J}{\partial u} = \begin{bmatrix} \partial J / \partial u_1 \\ \partial J / \partial u_2 \\ \vdots \\ \partial J / \partial u_{N_c} \end{bmatrix} \quad (5-26)$$

where u is the control vector and N_c represents the number of controls

Stochastic Gradient

Dealing with a deterministic optimization problem, the question arises: what is the advantage of using a stochastic gradient optimization approach in order to get the maximum of the objective function? Unfortunately for a practical large scale optimization problem, it is not possible to obtain the closed-form analytical solution of the objective function. In order to solve for one control vector u^* , that would maximize $J(u, x)$, an iterative algorithm is used. A step by step procedure is implemented that moves from a initial guess (or set of guesses) for u to a final value that is expected to be as close as possible to u^* (optimal control vector). Because in the set of all possible optimal control vectors U^* any one control vector is as good as any other it is common practice to seek only one u^* in the algorithmic search process. There is no reason to try and use an impossible amount of resources to find the full set U^* . One of the major difficulties in optimization is the distinction between global and local optimization, in our case global or local maximum. A global optimum solution to the maximization problem i.e. at least one u^* in the set U^* , with each $u^* \in U^*$ providing a higher value of J , than any other $u \notin U^*$. In most practical cases one must be content with obtaining a local solution because a global solution might not be available.

A stochastic gradient of $\bar{J}$ with respect to an ensemble of control vectors of size M , is obtained by performing M simulation of the reservoir model, each with a perturbed control vector u_i , ($i = 1, \dots, M$). The perturbed control vector is obtained from the initial control vector which is perturbed using a random vector sampled from Gaussian distribution with zero mean and standard deviation one. Thus in ensemble matrices the control and objective function standard deviation.

$$U = \begin{pmatrix} u_1^1 - \bar{u}_1 & \cdots & u_{N_u}^1 - \bar{u}_{N_u} \\ \vdots & \ddots & \vdots \\ u_1^M - \bar{u}_1 & \cdots & u_{N_u}^M - \bar{u}_{N_u} \end{pmatrix} \quad \text{and} \quad J_i = \begin{pmatrix} J_i^1 - \bar{J} \\ \vdots \\ J_i^M - \bar{J} \end{pmatrix} \quad (5-27)$$

where $\bar{u}_k = \frac{1}{M} \sum_{i=1}^M u_k^i$ for $k = (1, \dots, N_u)$, $J_i = J(u_i, x)$, $\bar{J} = \sum_{i=1}^M J_i$. Now applying linear regression through the M points $(u_i - \bar{u}, J_i - \bar{J})$ with $\bar{u} = (\bar{u}_1, \dots, \bar{u}_{N_u})^T$ results in a regression coefficient vector $\beta_i = (U^T U)^{-1} U^T J_i$ that approximates the gradient of J_i . Averaging the coefficients β_i leads to a stochastic gradient G that approximates the gradient of $\bar{J}$. Now the stochastic gradient G is defined as:

$$G = C_u^{-1} \frac{1}{M-1} \sum_{i=1}^M (u_i - \bar{u})(J_i - \bar{J}) \quad (5-28)$$

where $C_u \approx \frac{1}{M-1} (U^T U)$ and $\bar{J} = \frac{1}{M} \sum_{i=1}^M J_i$. Equation (5-28) requires the evaluation of the objective function M times, in the non deterministic case the computation of the stochastic gradient G would require M^2 evaluations of the objective function. This is due to the fact that for every member of the state vector ensemble of size M , the objective function is evaluated M times Olwijn et al.(2010). It was observed that a direct use of the stochastic gradient in a steepest ascent optimization procedure is not the best possible choice. To eliminate this lack of optimality a regularized gradient is obtained by pre-multiplication of G with a matrix R of the following form $R = C_u \cdot C_u$. It is suggested by Chen et al.(2009) this choice for R will provide a preconditioning for the steepest ascent optimization method as well as a smoothing to correct for noise resulting from the use of a small ensemble.

$$u_{i+1} = \frac{1}{\alpha} C_u G + u_i \quad (5-29)$$

Another interpretation, is that by premultiplying the gradient G with C_u , the smoothness that was lost while approximating G using a small-sized ensemble, is restored.

There are numerous studies that attempt to compute the gradient of the expected net present value with respect to a control variables. One way of doing exactly such a thing can be found in Sarma et al. (2005), where the adjoint method is used to compute the expectation of the net present value with respect to the controls, and the uncertainty of the reservoir description is propagated using the propagation collocation method. Other approaches also consider the use of NPV expectation as the objective function, they use an ensemble of reservoir models to reflect the uncertainty of the geological model and solve the adjoint equation for each realization. The average of the resulting gradients is considered to be the gradient of the expected NPV with respect to the control vector. Both approaches have one major drawback, that is they require the adjoint code to be available, which in most cases is something hard to obtain. Compared to these two methods EnOpt circumvents the need for the adjoint by simply approximating the average sensitivities of the expected NPV with respect to the control variables from the coupled ensemble of controls. It uses the reservoir simulator as a black box system, and it is fairly flexible in terms of choice of control variables and objective functions that need to be optimized Chen et al.(2008).

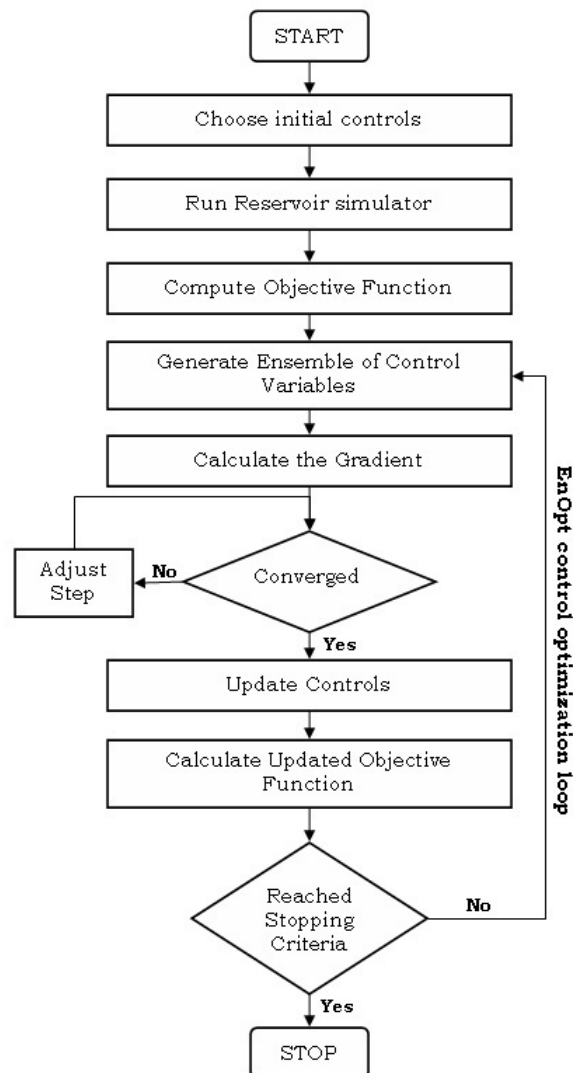


Figure 5-3: The flowchart for the Ensemble-Based Optimization

5-4-3 Implementation of ensemble-based optimization

The procedure of the ensemble-based production optimization implementation is summarized in Figure (5-3). A more detailed step by step view of the procedure is further presented. A general outlook contains three main parts, the initialization part, an outer loop and an inner loop. During the initialization phase all the required parameters of the procedure are declared, the underlying geological model is setup with the assumed rock and fluid properties. Also a background solution for the optimization problem is picked, in this case the background solution represents the control vector. Around the control vector an initial ensemble of control vectors is generated. Using this ensemble an initial search direction is calculated. The outer loop computes the objective function for the new controls and checks whether or not an improvement of the NPV is achieved. Also, a new set of ensembles of controls is generated and a new search direction is calculated. The inner loop provides further improvement of the NPV by adjusting the step size and/or the search direction. This is done using either line search or trust region.

EnOpt Initialization

Step 1 Let $i = 1$. Generate initial control variables u_1 and initial ensemble control variables $u_{1,j}, j = 1, 2 \dots N_e$. $u_{i,j}$ are generated in two steps. First, a mean control is sampled from a uniform distribution with upper and lower limits equal to the maximum and minimum possible well constraints for each realization of each well. Second, a temporally correlated Gaussian random field with zero mean is generated for each realization of each well and added to the mean control. Set $u_1 = \frac{1}{N_e} \sum_j = 1^{N_e} u_{1,j}$.

EnOpt Outerloop

Step 2 If $i \neq 1$ a temporally correlated Gaussian random field with zero mean is generated for each realization of each well and is added to the control variables u_i .

Step 3 Run the simulator for the entire chosen time interval and compute the objective function, with the help of the simulator results.

Step 4 Compute the gradient G which is also the search direction using equation 5-28.

Step 5 Compute the updated control variable u_{i+1} .

Step 6 Evaluate the objective function for the new controls $J(u_{i+1}, x)$, this requires N_e model simulations

Step 7 If $J(u_{i+1}, x) > J(u_i, x)$, then $u_i = u_{i+1}$ and let $i = i + 1$.

Step 8 If $J(u_{i+1}, x) < J(u_i, x)$ start the inner loop, keep the u_i decrease the step size by increasing α and go to Sept 5.

Step 9 Check if the stopping criteria has been reached. If not go to Step 5, else set $u = u_i$ and exit the outer loop.

Note that the stopping criteria that is checked in Step 9 usually refers to the relative increase of the NPV, meaning that if the increase of the objective function is lower than 0.01% then the stopping criteria is reached. Another option is to look at the change between two or more consecutive control vectors and if that change is less than 1% then again the optimization process is stopped. For testing purposes a stopping condition is not used. Instead a fixed number of iterations is used. This choice is supported by the fact that the optimization process has a high chance of reaching a local maximum instead of a global maximum. Given the stochastic nature of the method, using more iterations provides a better chance of escaping the local maximum and reaching a better local maximum or even a global one.

Numerical Results

This chapter contains all the numerical results of the model reduction approach to production optimization, described in Chapter 5. The method was implemented for two test cases and the results were compared with numerical results from EnOpt. The first experiment represents the implementation of MorOpt on a simple two-phase 2-D reservoir model and the second one represents the implementation of MorOpt on a more complex two-phase 3-D reservoir model. In both cases, the goal of the optimization process is maximization of the objective function. This chapter shows that optimizing the controls using MorOpt leads to a maximization in the NPV. All the numerical experiments were performed using an open source Matlab simulation toolbox (MRST-Sintef).

6-1 Matlab Reservoir Simulation Toolbox (MRST)

MRST represents an open-source Matlab toolbox that can be used as an efficient test platform for researching new optimization methods in reservoir simulations. The aim of MRST is to provide reproducible research and simplify the development, verification, validation and comparison of new methods on general grids. MRST contains a set of data structures and routines for creating, manipulating and visualizing petrophysical data, as well as routines for computing single and multiphase incompressible flow.

MRST has the following functionality for rapid prototyping of solvers for flow and transport:

Grids: MRST contains a common data structure and interface for all types of grids. It does not contains a grid generator but instead it contains grid factory routines for rectilinear grids, triangular and tetrahedral grids, 2D Voronoi grids,extrusions of areal grids to 2.5D volumetric grids. It also contains numerous tutorial examples and a few realistic data sets.

Input and Output: MRST provides routines for reading and processing industry standard input files for grids, petrophysical parameters, fluid models, wells, boundary conditions, and simulation setup.

Parameters: The toolbox works with a data structure for petrophysical parameters and a few, very simplified, geological routines. The version of MRST used, 2011a supports just incompressible fluid models, but a version that supports compressible black-oil fluids has been announced. Also as parameters of the simulator, we could consider routines for setting and manipulating boundary conditions, sources/sinks and well models.

Units: MRST works in a strict SI units but supports conversion to and from other unit systems like field-units. The simulator supports multiple types of unites, when an industry-standard input format is not used, the user is responsible for the explicit conversion and consistency of the units.

Reservoir state: The toolbox works with results data structure that contains state variables like pressure saturation and fluxes in every grid cell of the reservoir.

Postprocessing: MRST has implemented visualization routines for scalar cell and face data, that make plotting useful results very easy.

Solvers: The toolbox contains several flow and transport solvers which may be readily combined using an operator splitting framework. In particular it provides an implementation of the multiscale mixed finite-element method working on unstructured, polyhedral grids.

Linear algebra: MRST relies on Matlab's build in linear solvers but these can easily be replaced by specialised solvers.

6-2 Numerical Results 2D Case

For the numerical experiments presented in this section, the model reduction approach is applied in order to get an optimized control vector. The optimized control vector provides a maximization of the NPV. First the simple 2D study case is chosen and all the data from the numerical experiments is recorded and analysed. When the more complex 3D study case is considered, the results from it are expected to be similar to the 2D study case. No significant differences are expected to be observed between the two numerical trends.

Model Settings

The model used for the numerical experiments is a simple synthetic 2D reservoir model, which describes iso-thermal incompressible two phase (oil and water) flow, see Figure (6-1). All the numerical experiments were performed using the MRST reservoir simulator presented in beginning of this chapter. All the model parameters are assumed to be known including permeability and porosity. No history matching

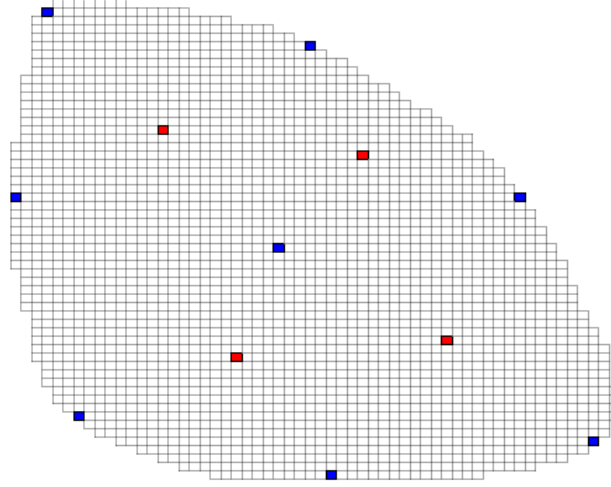


Figure 6-1: The reservoir Grid used in this section for the numerical experiments. The red squares correspond to the producers and the blue squares correspond to the injectors

is done for the reservoir. The 2D reservoir model represents the first layer of the 3D model used in the next section.

Model assumptions: The reservoir operates under water flooding conditions. Given the fact that the simulator only supports incompressible flow, this becomes a forced model assumption. Also, it is assumed that the effect of the gravity can be neglected, the permeability is known and that all parameters are pressure independent. Dealing with a two-phase oil-water problem, first the two-phase pressure equation is solved:

$$\nabla \cdot v = q, \quad v = -\lambda K \nabla p, \quad (6-1)$$

where v is the Darcy velocity (total velocity) and λ is the mobility, which depends on the water saturation S . The saturation equation (conservation of the water phase) is given as:

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (f_w(S)v) = q_w \quad (6-2)$$

where ϕ is the rock porosity, f is the Buckley-Leverett fractional flow function, and q_w is the water source term.

Fluid properties: As previously mentioned we are dealing with a two-phased problem, for which the relative permeabilities Figure (6-2), represents the additional resistance to flow of a phase. This is caused by the presence of the other phase and follows a Corey model. Corey exponents for oil and water are equal to two, and the

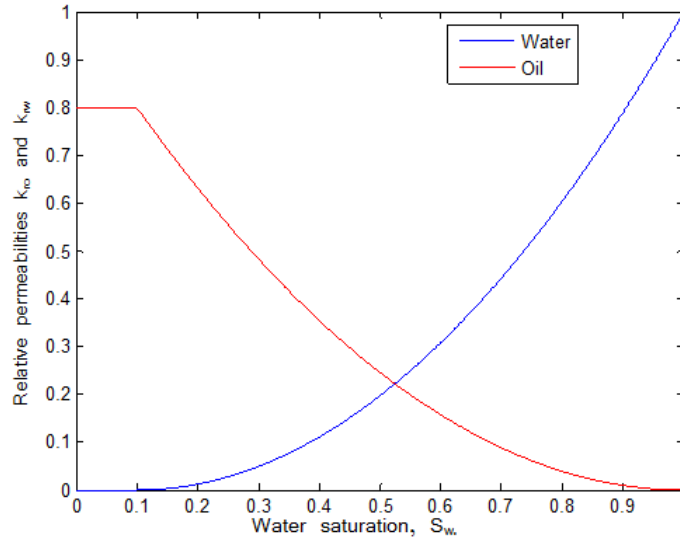


Figure 6-2: Relative permeability curves.

end points for the relative permeabilities of oil and water were taken equal to 1 and 0.8, respectively. The residual oil saturation and the connate water saturation were considered to be equal to 0.2. Oil and water viscosity was chosen as $1 \cdot 10^{-3} Pa \cdot s$ water and $5 \cdot 10^{-3} Pa \cdot s$ oil.

Reservoir geometry and properties: The reservoir is constructed as a Cartesian grid of size 60-by-60-by-1 cells, where each cell has dimension 10-by-10-by-10 meters. The permeability K is set to be homogeneous, isotropic and equal 100 mD and the porosity to be equal to 0.3 for all the grid cells. Not all cells are considered active, only 1853 cell are active and give the egg shape from Figure (6-1).

Initial conditions: The initial reservoir pressure is assumed to be $32.5 \cdot 10^6 Pa$ and the initial water saturation is taken as the connate water saturation.

Well settings: There are 12 wells, eight injector wells and four producer wells, positioned as shown in Figure (6-1). The injector well is a vertical well controlled by injection rate, which is going to be optimized in order to get an improvement in the NPV. The four producer wells are also vertical wells, but are controlled by bottom hole pressure (bhp) this is set at $30 \cdot 10^6 Pa$. MRST uses Peacemann model in order to describe the wells, this gives an extra set of equations, that need to be assembled. For a vertical well, the model is of the following form:

$$q = J(p - p_{bh}), \quad J = \frac{2\pi Kh}{\mu(\ln(r_e/r_w) + s)}, \quad r_e = 0.2\sqrt{\Delta x \Delta y} \quad (6-3)$$

where q is the flow rate p_{bh} is the bottom-hole pressure, r_e is the effective (Peaceman) radius, r_w is the radius of the well, K is the permeability, and h the height of the well. The radius of the wells was set 0.1 meters.

6-2-1 Production optimization settings

All the numerical optimization experiments were performed, under the same settings: injectors are only controlled by water injection rate, additionally there is a constraint on the total injection rate which is, given the small size of the 2D case, chosen to be $10 \text{ m}^3/\text{day}$. This very low maximum constraint on the total injection rate and was chosen because of two decisive factors. First the relative small size of the reservoir and volume of liquid it contains ($1.2754 \cdot 10^5 \text{ m}^3$) and second the reservoir simulator works only with incompressible flows. This combined with the fact there is no boundary flow, leads to the observation that the volume of liquid injected is equal with the volume of liquid that is observed from the producers.

In order to evaluate the NPV, for our optimization problem we need to specify some financial parameters. The price of oil is set as 150\$ the cost of water disposal is set at 25\$ and the cost of water injection is set at 5\$. The discount rate is chosen to be 8 %. Time frame for the NPV optimization is four years, 1460 days, the control settings are modified every 182 days, which gives us eight control steps.

The 2D study case contains eight injectors and the total number of controls is equal with the number of control parameters times the number of wells, i.e. 64 control variables. At every control time every one of the eight wells has its injection rate updated. The ensemble used for constructing the snapshots contains 25 members and a complete optimization process involves 15 runs of the outer loop. An initial control vector is chosen and is updated for every control step so that we obtain an improvement of the NPV. In order to get the updates for the control vector the reduced order model is constructed, and the controls are optimized in the reduced space. This involves generating the 25 members of the ensemble by perturbing the initial controls with a Gaussian noise with mean zero and standard deviation 0.05. Using the generated ensemble by running the full order model for each member in the ensemble, the snapshot matrix is constructed. From the snapshot matrix, the main patterns that are used in constructing the reduced order model, are obtained using the POD technique. In the end we compare the results from the constant injection rate approach to water flooding with the results from the optimized controls, for both the MorOpt and EnOpt.

6-2-2 Production optimization results

In production optimization the goal is to find the best strategy which will maximize an objective function which can either be the net present value (NPV) or the ultimate recovery (UR). This section will show that MorOpt can do exactly this in a short computational time and in a robust manner. Here robustness is defined as the ability of the method to provide similar maximum optimized values for the

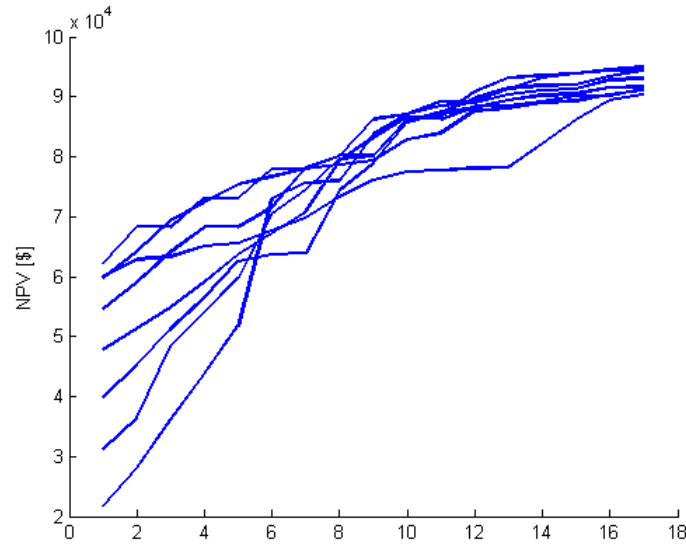


Figure 6-3: NPV for the case when the optimization process is started using different initail controls

NPV independent of the initial control vector chosen. Also the optimization results obtained using the MorOpt are compared with best case scenario optimization results obtained using EnOpt. The EnOpt process can contain two iterative search methods that describe how the optimization process goes from one iterate to the next iterate. This methods are called linesearch and trustregion. In this section when referring to EnOpt we assume that the method uses trustregion in order to provide the optimized control strategy. The results, of EnOpt using linesearch are not presented in this section because, they prove to be similar to the results obtained from EnOpt using trustregion. The method robustness is evaluated by starting the optimization process from different initial control and checking if the optimization process converges to the same level NPV toward the end, Figure (6-3). This test shows that the final result of the optimization process is independent of the initial chosen control vector.

This is a good thing, because it shows that unlike using a simple implementation of the steepest ascent method, using MorOpt it is more likely to reach a optimum which is global than local.

Table (6-1) presents the same information as Figure (6-3) as well as the NPV improvement, which is computed as the percentage the NPV is higher than the NPV obtained for initial constant controls.

Pressure and saturation have different physical behaviour and also have a different variability in their values, therefore they were considered separately when the patterns were chosen. The pressure behaviour shows less variability than the saturation behaviour. It is expected that more patterns would be required to represent the saturation field than the pressure field. As previously mentioned, because of

Case	Injection Rate [m^3]	Initial NPV [$10^4\$$]	Optimized NPV [$10^4\$$]	NPV Increase [%]
1	64	5.9962	9.0382	50.73
2	96	6.2157	9.1335	46.94
3	128	5.9755	9.1075	52.41
4	160	5.4633	9.4940	73.77
5	192	4.7853	9.4490	97.46
6	224	3.9919	9.4390	136.45
7	256	3.1147	9.1662	194.28
8	288	2.1749	9.3084	327.99

Table 6-1: Optimization results for different background controls

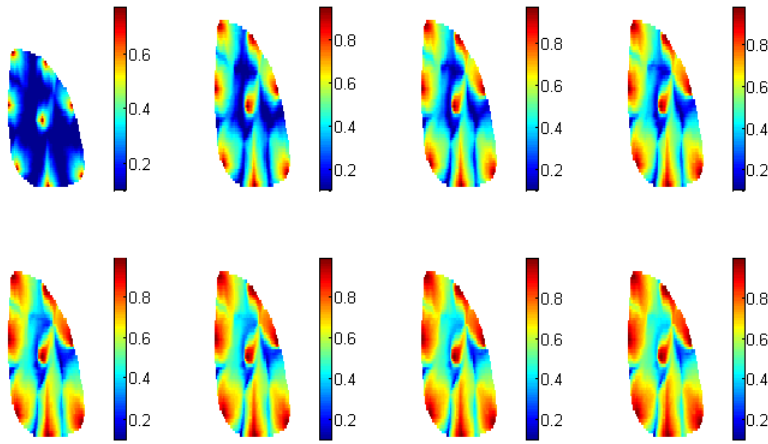


Figure 6-4: Water saturation distributions at six month time interval

the limitation of the reservoir simulator in dealing with compressible flow the pressure patterns are ignored and the reduced order model is constructed using only saturation patterns. In order to cover the dynamic behaviour of the system, snapshots were taken during all the model integration times for every member in the control ensemble. Figure (6-4) describes the behaviour of the saturation field under water-flooding condition for the 2D test case. Initially the reservoir is completely saturated with oil (blue), then water (red) is pumped through the eight injection wells.

A number of main parameters can be identified as being important for the performance of the model reduction technique: size of the control ensemble, standard deviation used when the ensemble is generated, perturbation size of the finite difference approximation, relative importance used when picking the saturation patterns and the control patterns. For all this parameters a sensitivity analysis is done, using the 2D model, Table (6-2). The first case from Tabel (6-2) represents the reference case parameters that is seen as a blue line in Figure (6-5).

The information present in Table (6-2) can also be visualized in Figure (6-5). From

Case	Rel Importance	Nr of Ensemble Members	STD	Pert Size	NPV [\$]	# Model Runs
1	0.7	25	0.05	10^{-4}	9.1795	357
2	0.7	25	0.1	10^{-4}	9.1512	384
3	0.7	25	0.2	10^{-4}	8.2414	320
4	0.7	50	0.05	10^{-4}	9.2021	598
5	0.7	100	0.05	10^{-4}	9.4846	1163
6	0.8	25	0.05	10^{-4}	8.4846	423
7	0.7	25	0.05	10^{-3}	6.7766	492

Table 6-2: Optimization results for different parameter values

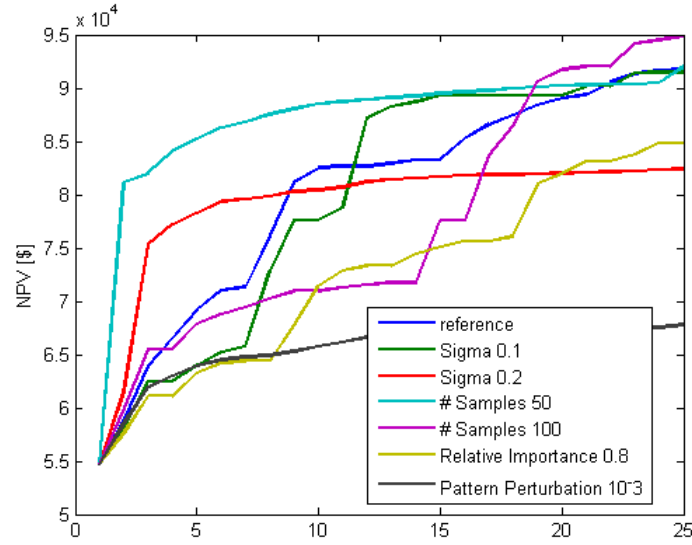


Figure 6-5: MorOpt Parameter Sensitivity

the results present in Figure(6-5) and Table (6-2) it is clear that using a total relative importance of 70% is enough to capture the main dynamics of the small 2D reservoir model. Also a small standard deviation is better when generating the control ensemble. This is due to the fact that our reduced order model is accurate in the vicinity of the background controls and too much variation in the control ensemble has a negative effect on the final results of the process. On the other hand increasing the number of samples also increases the maximum NPV value reached by the optimization procedure. This in turn also is positively correlated with the full order model runs that the optimization process requires.

The comparison of the optimization results, coming from MorOpt using reparametrized controls and the best case parameters, which are: relative importance 70% 100 ensemble members used when generating the snapshots for the reduced order model, 0.05 standard deviation and 10^{-4} perturbation size for both the control patterns and the saturation patterns. With the optimization results coming from the EnOpt method, again using a set of optimized parameters, and with MorOpt using Non-reparametrized controls is present in Figure (6-6).

MorOpt using non-reparametrized controls builds the low order model using the saturation patterns and the sensitivity of the model with respect to the controls. The sensitivity is approximated using finite difference for every element in the control vector. Concluding that every time the reduced order model is build, we need to run the full order model $N_u + p$ times, where N_u is the number of elements in the control vector and p is number of saturation patterns. This is not a particular problem for small test cases, but as the test case grows, more injection wells, more control times, resulting in more control elements in the control vector. This also leads to more full order model runs in order to build the reduced model, which definitely cuts the computational advantage that the method is supposed to provide. Never the less, Figure (6-6) shows the MorOpt using non-reparametrized controls performs better in terms of the maximum NPV the optimization process reaches, when compared with the MorOpt using reparametrized controls. But this is done with a clear computational disadvantage when the number of full model runs is analysed. The same thing is true for EnOpt, Figure (6-6), shows that EnOpt performs better than both the MorOpt using non-reparametrized controls and MorOpt using reparametrized controls but at a higher computational cost than both of them. This information is better visualized in Table (6-3). Figure (6-6), suggest that the computational cost observed from EnOpt can be reduced by implementing an stricter stopping criterion but this is also true for other two method as well. For the purpose of this thesis, a strict stopping criterion was not implemented, in the case of EnOpt, due to the stochastic nature of the gradient obtained using the method. Details were presented in Chapter 5.

Case	Optimization Method	Max NPV Reached [$10^4\$$]	Number of Model Runs
1	MorOpt Reparametrized Controls	0.9494	361
2	EnOpt	1.0242	2638
3	MorOpt Non-Reparametrized Controls	1.0047	870

Table 6-3: Comparison of Optimization Results 2D case

The number of full order model runs from Table (6-3), shows that, MorOpt using re-parametrized controls, provides a 86.31 % reduction in number of model runs, which can be directly translated to a reduction in overall computational time, while providing a maximum NPV which is 7.3% smaller than the maximum NPV obtained using EnOpt. Compared to MorOpt using Non-reparametrized controls, MorOpt using reparametrized controls provides a 58.5 % reduction in number of model runs, while providing a maximum NPV which is 5.5% smaller than the maximum NPV obtained using MorOpt using Non-reparametrized controls.

The production strategy for the three methods, Figure (6-7) shows the optimized controls that provide the maximum NPV from each of the three methods. There is an upper and lower bound for the injection rate, which is put in place so that the reservoir simulator does not return non-physical values for pressure. The initial control vector contains constant injection rates for every injector at every time step. There is no correlation set on any given two controls at consecutive time steps, i.e.

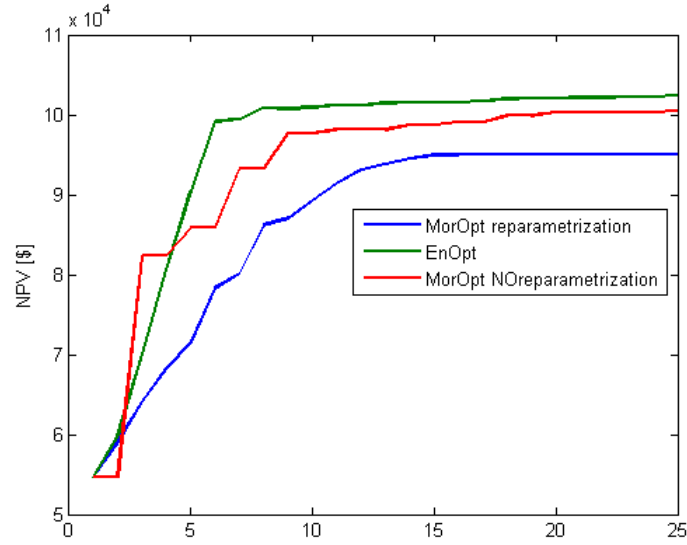


Figure 6-6: 2D Comparison of Optimization Results

the optimized control vector can contain the maximum value for the injection rate of one injector at a specific time, and for the next time step the minimum value for the same injector. Figure (6-7) presents the optimized control strategy returned by the three optimization methods, MorOpt with reparametrized controls (blue line), EnOpt (green line) and MorOpt with Non-reparametrized controls (red line). It shows that the optimized control strategy of the three methods is similar for the injection wells.

Figure (6-8) describes the evolution of the total volume of water, in m^3/day , observed in the four producing wells of the reservoir. The water volumes come from the optimized controls of the three optimization methods, MorOpt with reparametrized controls (blue line), EnOpt (green line), MorOpt without reparametrized controls (red line). Because the NPV objective function penalises water production and injection and rewards oil production, with different weights. It is interesting to observe the optimization method that provides optimized controls which deliver the lowest water production rates. Looking at Figure (6-8) it is not easy to decide which of the three methods does this. The decision can be simplified by considering Figure (6-9) where the water production rates for all the producers are summed together and presented as one.

One other indicator of performance of the optimization process could be the moment in time, when water is observed in the production wells, also more commonly known as water breakthrough time. Although, the optimization methods were not designed to increase the water breakthrough time it is worth mentioning that Figure (6-9) shows, that for all the three optimization methods, water breakthrough is present after a period of six months.

Figure (6-10) describes the evolution of the total volume of oil, in m^3/day , observed in the four producing wells of the reservoir. Again, oil volumes come from the

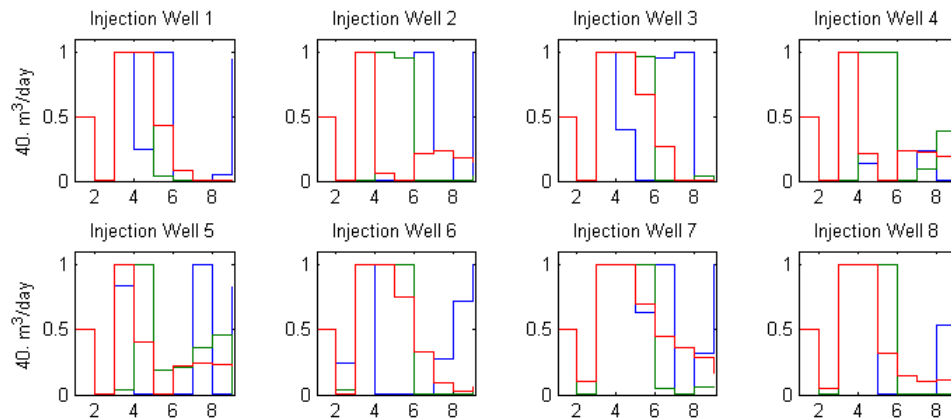


Figure 6-7: 2D Optimized Control Strategy

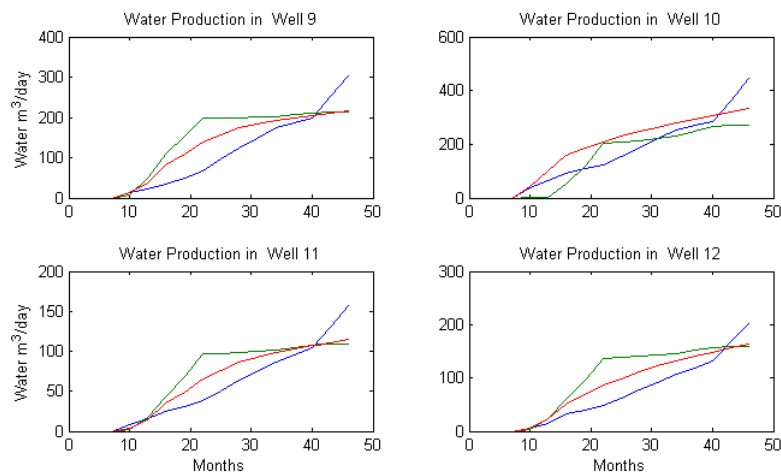


Figure 6-8: Cumulative Water Production for each Production Well

optimized controls of the three optimization methods, MorOpt with reparametrized controls (blue line), EnOpt (green line), MorOpt without reparametrized controls (red line). Again, it is not easy to observe which of the three optimization methods provides optimized controls that results in the highest oil rates. Figure (6-11) makes is more clear by combining the overall oil production into one picture.

Figure (6-11) and Figure (6-9) show that the optimized controls obtained using MorOpt with reparametrized controls leads to the highest volume of produced oil but also to the highest volume of produced water. Also, by taking into account that the difference in oil volume production between the three methods is not that significant, it is now clear where the difference in the maximum NPV comes from. For more details about the optimized water and oil volumes in Figures (6-9) and (6-11) see Table (6-4).

The results shown by Table (6-4) are not surprising, because as previously men-

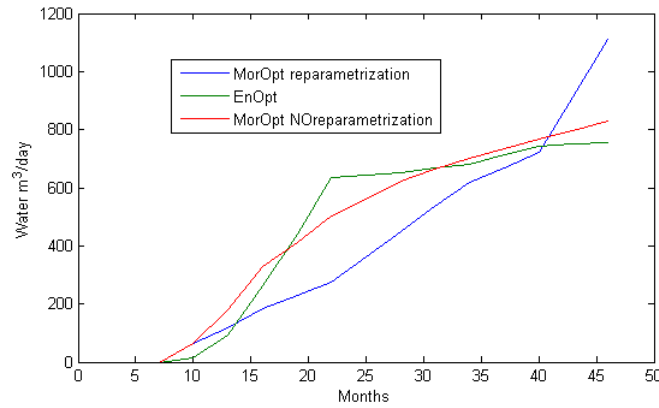


Figure 6-9: Cumulative Water Production

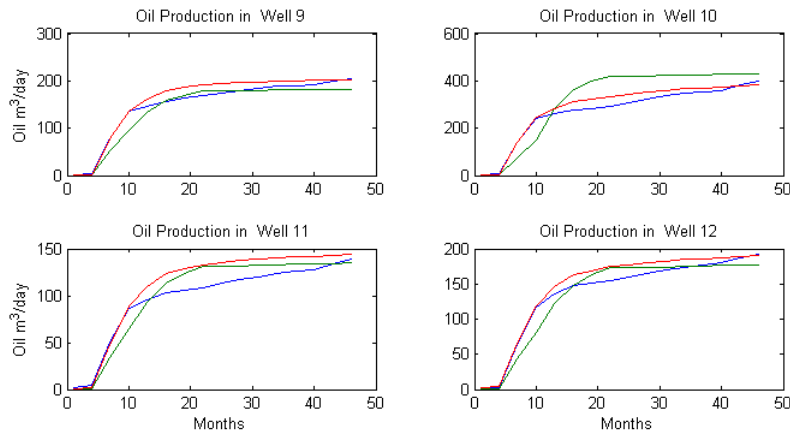


Figure 6-10: Cumulative Oil Production for each Production Well

tioned in the beginning of this chapter MRST, the simulator used for the numerical experiments, does not support compressible flow. This fact coupled with no flow boundary conditions, means that the amount of liquid, for the purpose of this thesis, water injected in the reservoir. Is equal with the amount of liquid observed as production from the producer wells. Although the intuition is correct, in the sense that higher volumes of oil translate to higher NPV values. The explanation for the higher values of the NPV in the optimized cases, observed for a lower volume of oil recovered, comes from the properties of the objective function. As previously described, the NPV function rewards oil production and penalizes water injection and water production. In Table (6-4), even though we observe higher volumes of produced oil, we also observe higher volumes of water production and higher volume of water injected over the lifetime of the reservoir.

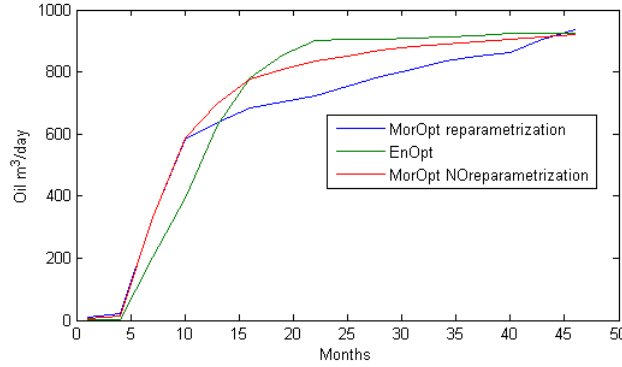


Figure 6-11: Cumulative Oil Production

Case	Optimization Method	Inj Water [$10^3 \cdot m^3$]	Water Prod [$10^3 \cdot m^3$]	Oil Prod [m^3]	NPV [$10^4 \$$]
1	MorOpt Repar Cont	2.0470	1.1094	937.6139	0.9494
2	EnOpt	1.6815	0.7552	926.3395	1.0242
3	MorOpt Non-Repar Cont	1.7475	0.8278	919.7572	1.0047

Table 6-4: Comparison of Total Liquid Volumes for 2D case

6-2-3 Conclusions

In the experiments presented in this chapter the model reduced production optimization method was used in order to obtain a control strategy that maximized the NPV, as the objective function. The method was further refined by reparametrizing the control space and was compared with an ensemble based optimization method and with MorOpt without reparametrized controls. The following conclusions were drawn from this study:

- The presented POD-based approach is adjoint free, is relatively easy to implement and can be implemented around any reservoir simulator.
- MRST as the reservoir simulator used for the numerical experiments, is a Matlab toolbox that can be used as an efficient test platform for researching new optimization methods. Even though it proves to be a versatile and easy to work with. We need to consider also the limitations it poses, such as, the fact that it simulates only incompressible flows.
- First thing that was tested is whether or not by reparametrizing the control space we could still get reasonable optimization results. This is confirmed by Figure (6-3), where it is clear that by starting from different background controls, we manage to converge to a similar value for the maximum NPV.
- We looked at the development of the NPV and the control strategies obtained by changing key parameters of the model-order reduction method. It was useful for the purpose of observing if MorOpt method was overly sensitive to certain parameters and also made it possible for us to pick the combination of parameters which provides the biggest increase in the final NPV.

- From the several tests performed these parameters, we observed that their impact on the optimization process to be significant. The size of the ensemble used to generate the snapshots is a relevant parameter. Because the reduced order model is a linearized form of the original equations, the snapshots should be taken in a relative close vicinity of the linearized trajectory. Choosing less snapshots and less patterns results in less variability and may results in the optimization process converging to a maximum NPV which is significantly lower than the optimized NPV obtained using EnOpt.
- The numerical experiments show that a relative importance of around 70% is sufficient in order to capture the significant dynamics of the full order model using the reduced order approximation. This percentage for the total relative importance is expected to increase with the increase of the complexity of the reservoir. The number of snapshots used, the total relative importance used and number of patterns used, all have a direct proportional relationship with the computational cost of the reduced-order method.
- In both cases with and without reparametrized controls MorOpt performed very well in terms of the maximum NPV value reached and also with respect to the number of full order model runs needed, in order to complete the optimization process.
- MorOpt using re-parametrized controls, provides a 86.31 % reduction in number of model runs, while providing a maximum NPV which is 7.3% smaller than the maximum NPV obtained using EnOpt. Compared to MorOpt using Non-reparametrized controls, MorOpt using reparametrized controls provides a 58.5 % reduction in number of model runs, while providing a maximum NPV which is 5.5% smaller than the maximum NPV obtained using MorOpt using Non-reparametrized controls.

Though all this conclusions are true for this rather small 2D model, they are expected to have a small variation when dealing with a more complex 3D model.

6-3 Numerical Results 3D Study Case

Continuing the study started on the 2D model, the MorOpt method is used to perform production optimization on a more complex 3D model. Like in the 2D case the control vector is optimized in order to obtain the best production strategy that maximizes the objective function. Asadollahi et al. (2009) shows that the well liquid rates are the variables that present the strongest impact while trying to maximizing the NPV using a gradient based optimization method. In other works well bottomhole pressure has also been suggested to be a significant factor in influencing the maximum NPV the process reaches. Based on Asadollahi work, for the purpose of this thesis the control variables in the control vector, that we are trying to optimized, represent the water injection rates.

Model Settings

The model that is used for the numerical experiments, was first introduced by Van Essen et al.(2009) and describes iso-thermal incompressible two-phase oil water flow, in a channelized reservoir with eight injection and four production wells, see Figure (6-12) for details. The model is based on the assumption that the effect of capillary pressure and gravity can be neglected and the permeability is isotropic.

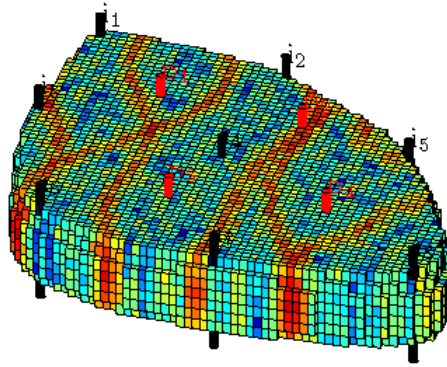


Figure 6-12: Reservoir permeability field with well locations. Injectors are indicated in black, producers in red

Reservoir geometry: The reservoir model is part of a rectangular parallelepiped of the size $480m \times 480m \times 28m$, divided into $60 \times 60 \times 7$ uniform Cartesian grid blocks, of which there are 18553 active cells, forming an egg-shaped domain with no flow boundaries.

Reservoir properties: The porosity is assumed to be uniform and equal to 0.2. The "true" permeability field on a log scale can be seen in Figure (6-12). The permeability contrast between the channels and the background fill is relatively mild.

Initial conditions: Initial reservoir pressure is set at 400 barsa ($40M \cdot Pa$), the bottom hole pressure in the production wells is set at 395 barsa ($39.5M \cdot Pa$) and the initial water saturation is taken as the connate water saturation $S_{wc} = 0.2$

Well locations and constraints: The optimization process starts from a constant injection rate of $140m^3/day$ for each of the eight injection wells, totalling at $1120m^3/day$

for the entire reservoir. A maximum possible injection rate is set at $480m^3/day$ for each injector. Meaning that at no time during the optimization process, the water injection rate for any injection well can exceed $460m^3/day$. The position of the injection and producers can be observed in Figure (6-12). All the wells are perforated at each layer of the reservoir and no constraints are imposed on the evolution of the controls. The upper limit set for the water injection volume is calculated from the pore volume of the reservoir. It is considered that the initial volume in the reservoir is represented by the pore volume and in order to completely exchange that volume with water we would need to inject a constant $3840m^3$ of water every day.

Fluid properties: The oil and water viscosities are identical: $\mu_o = \mu_w = 10^{-3}Pa \cdot s$. The density of oil and water are given by $\rho_o = 900kg/m^3$, and $\rho_w = 1000kg/m^3$.

Relative permeabilities: We assumed Corey-type relative permeabilities with Corey exponents for oil and water $n_o = 4$ and $n_w = 3$. The residual oil saturation was taken as $S_{or} = 0.15$.

Reservoir simulator: Like for the 2D model numerical experiments, MRST reservoir simulator has been used. As previously mentioned the simulator has a build in assumption, which is: incompressible flow. Other assumption made for the purpose of the numerical simulations are, no effect of capillary pressure and no gravity effect.

Production optimization settings

The numerical optimization experiments were performed, under the following settings: The injectors are only controlled by water injection rate. As in the 2D case the upper limit constraints defined for the injection rate, are in place in order to provide physically realistic values for pressure and saturations. In order to evaluate the NPV, some financial parameters need to be specified. The price of oil is set as 150\$ the cost of water disposal is set at 25\$ and the cost of water injection is set at 5\$. The discount rate is chosen to be 8 %. Time frame for the NPV optimization is four years, 1460 days, the control settings are modified every 182 days, which gives us 8 as the number of control steps. The 3D study case contains eight injectors, then the total number of control is equal with 64. This is the total number of wells multiplied with the number of control times. For every control time we have eight wells for which we update the control vector. The ensemble size used for generating the snapshots is chosen to be 25. The same ensemble size is chosen for EnOpt. Even though the 2D numerical experiments suggest that for both the MorOpt and EnOpt a large ensemble size leads to better optimization results. We choose to limit the number to 25 because a large ensemble size leads to a large number of full model runs, hence a high computational time. For as little as 25 ensemble members the optimization methods will need around 600 full order model runs to finish the process, and for the 3D reservoir model this means around 25 hours for one optimization test.

Even though initial test for the 2D for MorOpt with reparametrized controls, suggest that 70% relative importance for both saturation patterns and control patterns, will results in enough patterns, that are able to capture the desired full order model dynamics. When the same relative information is used for the 3D numerical tests, it is clear that the optimization process provides optimized controls that do not improve the initial NPV to the same order of magnitude that EnOpt does. This is due to the fact that reduced order model, in the case of the 3D reservoir model needs to contain more information about the sensitivity of the full order model with respect to the controls. This information can not be obtained using just 70 % relative importance. Also from the numerical test we see that the method performs better when different relative importance levels are chosen for saturation and controls. The best optimization results are obtained for 70% relative importance for saturation and 95-99% relative importance for controls. 95% relative importance was chosen for the controls because a 99% relative importance does not provide significant increase in the maximum NPV but significantly increases the number of control patterns, hence also, the computation time of the method. In the case of MorOpt without reparametrized controls a 95% relative importance is chosen for saturation. This is due to the fact that only saturation patterns and the sensitivity of the full order model with respect to each individual control are needed in order to build the reduced order model.

Production optimization results

Unlike the 2D numerical experiments, when comparing optimization results from MorOpt with and without reparametrized controls, with results from EnOpt, we will also include results from EnOpt using linesearch. Even though from Figure (6-13) it is easy to see that EnOpt using linesearch has a tendency to get stuck in a local maximum.

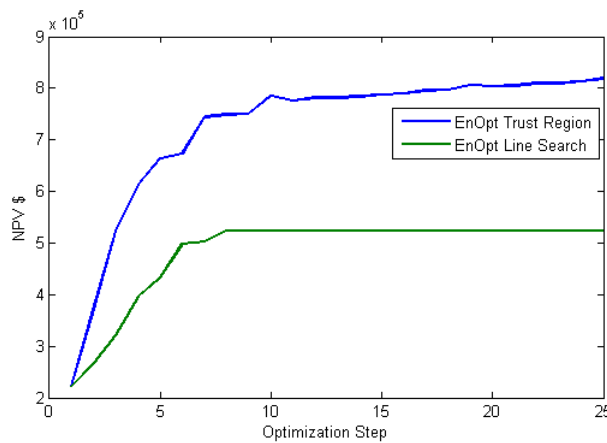


Figure 6-13: Comparative Optimization Results

All the numerical experiments start with the same control vector, which describes a constant injection rate of $140m^3/day$ for each injection well or $1120m^3/day$ for

the whole reservoir. This amounts to half of the maximum possible injection rate. Looking at Figure (6-13) it is clear what influence picking a certain iterative method, line search or trust region, has on the optimization of the NPV. It is clear that, even though for the 2D numerical experiment there was no significant difference. For the 3D case there is a quite significant difference in the maximum EnOpt reaches. This difference depends on what iterative method we chose to use. EnOpt using trust region has a stepper increase in the NPV for a lower number of iterations and also reaches a maximum value which is greater than the value reached by EnOpt using line search. On the other hand it does not appear to converge to a maximum value and it ends in an ascendant trend. This suggests that if more iteration are allowed the NPV function might increase even more. EnOpt linesearch appears to get stuck in a local maximum after just a few iterations of the optimization process. As in the 2D case there is no convergence criterion implemented for the method in order to benefit from the stochastic nature of the gradient.

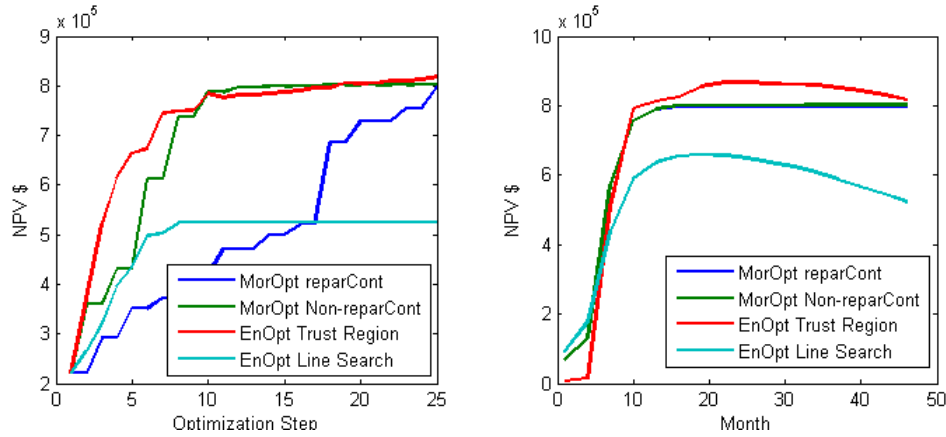


Figure 6-14: EnOpt two iterative methods

Figure (6-14), presents in the left image the comparison between the two optimization methods, MorOpt with and without reparametrized controls, and EnOpt with trust region and with line search. The left image shows how the objective function develops during the optimization process and the right image shows the evolution of the NPV under optimized controls provided by the two methods. Table (6-5) presents more details from Figure (6-14) that cannot be seen by just analysing the picture, it also includes the computational cost of each approach.

Case	Optimization Method	Max NPV [10^5 \$]	Model Runs
1	MorOpt Repar Cont	7.9937	453
2	MorOpt Non-Repar Cont	8.0294	736
3	EnOpt Trust Region	8.1718	686
4	EnOpt Line Search	5.4475	712

Table 6-5: Comparison of Optimization Results 3D Case

It is easy to see that EnOpt using trust region provides a control strategy that results in the highest NPV. Also comparing the computational cost of the method we

can see that in this particular example it is more cost effective than using MorOpt without reparametrized controls. MorOpt with reparametrized controls performs better in terms of maximum NPV reached by the optimize controls when compared with EnOpt using line search, but not better than the other two approaches. In terms of computational cost, MorOpt with reparametrized controls is better than the other three approaches, it is 38.45% less computational demanding than MorOpt without reparametrized controls reaching an maximum NPV which is 99.55% of the NPV reach by MorOpt using non-reparametrized controls. It is 33.96% less computational demanding than EnOpt using trust region reaching 97.82% of the maximum NPV reached by EnOpt using trust region. As it can be seen from Figure (6-13) and Figure (6-14) EnOpt using line search does not appear provide a control strategy that can reach the same level of NPV as the other three approaches. This might be due to the fact that for the line search approach a search step is defined and the size of the search step needs to be optimized during the optimization process. The step size can be optimized by considering it one more variable in the control vector. This information was not available during the implementation of the EnOpt and because of this is to integrated in the code.

Figure (6-15) presents the optimized production strategy obtained using one of the four mentioned approached, MorOpt with reparametrized controls (blue line), MorOpt without reparametrized controls (green line), EnOpt using trust region (red line) and EnOpt using line search (light blue line).

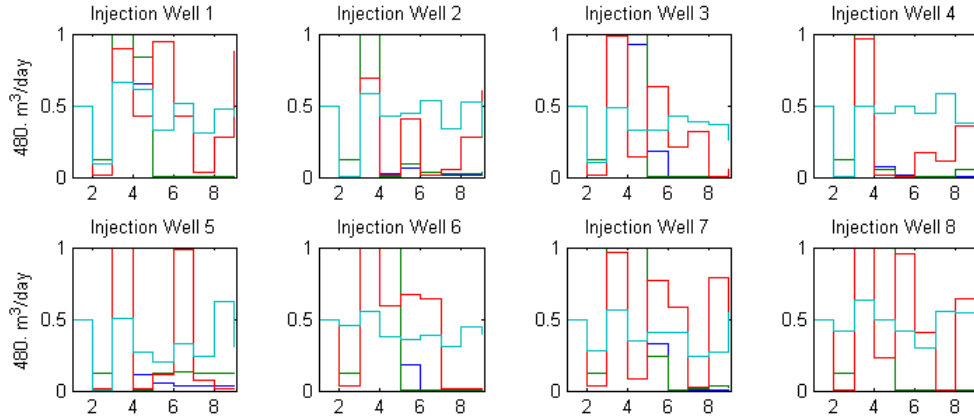


Figure 6-15: Comparative Production Strategy

Looking more closely at the optimized strategy provided by EnOpt using line search, it is easy to see that, the optimized control strategy elements never reach the maximum injection possible or are completely shut down. This suggest that the optimization process has not stopped and further improvements of the objective function can be obtained. This is shown to be possible by the other three approaches.

Figure (6-16) presents the total volume of water that is injected in the reservoir using the optimized controls from the four optimization approaches. It is easy to observe that EnOpt both with line search and trust region present a higher volume

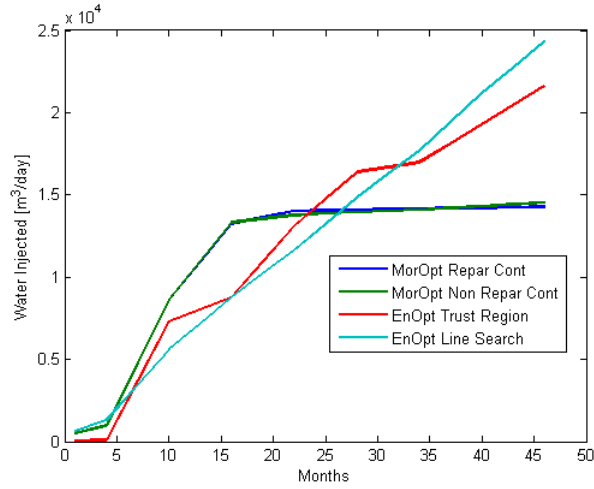


Figure 6-16: Optimized Water Injection Strategy

of water injection, than MorOpt with and without reparametrized controls. Taking into account the fact that the reservoir simulator works under incompressible flow. We conclude that using the optimized controls from the EnOpt method will result in a higher volume of liquid that is observed in the four producers.

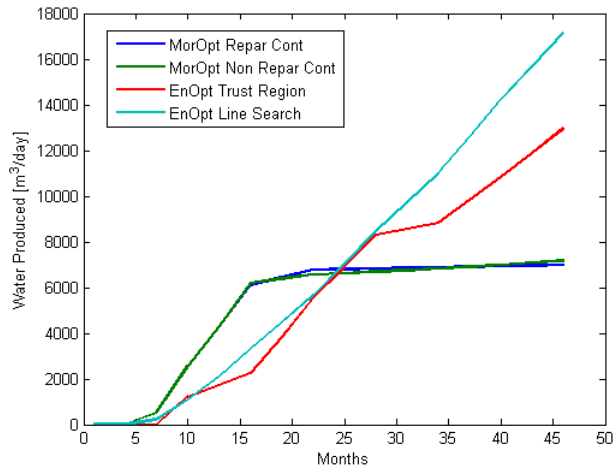


Figure 6-17: Water Production Under Optimized Controls

Higher liquid volumes produced are not always what is needed. One such example would be the case when the in total volume of liquid produced, more water is observed than oil. This will result in a low NPV value. For this purpose Figure (6-17) and Figure (6-18) present the total volume of water produced and the total volume of Oil produced. From these two figures and also from Figure (6-16) it can also be seen where the difference in the maximum objective function reached using optimized controls, comes from. Figure (6-17), similar to Figure (6-16), shows that using ensemble based optimized controls, water production volumes that are

higher than those coming from MorOpt optimized controls, are reached. This should result in a low value for the objective function which penalizes water production and injection with different weights. This is not the case for EnOpt using trust region as Table (6-5) shows us. Looking at Figure (6-18) we can see where the difference in the max NPV value comes from. Using the optimized controls from EnOpt trust region, higher volumes of water injected and produced are observed, also significantly higher volumes of oil are produced. Furthermore, the impact attributed to water production and injection and also oil production are different. The volume of oil produced has a bigger positive impact on the final NPV value and the water injection and production volumes have a smaller negative impact on the same final NPV value.

For the MorOpt method, we can see that the two approaches, using reparametrized controls and non-reparametrized controls, provides a very similar control strategy that results in similar volumes of water injection and production as well as oil production. This explains why the maximum NPV value reached using the optimized controls from both approaches are so similar. Using the optimized strategy provided by the EnOpt line search approach, we detect high volumes of water injected and produced but lower levels of oil volumes produced, hence, the low final NPV.

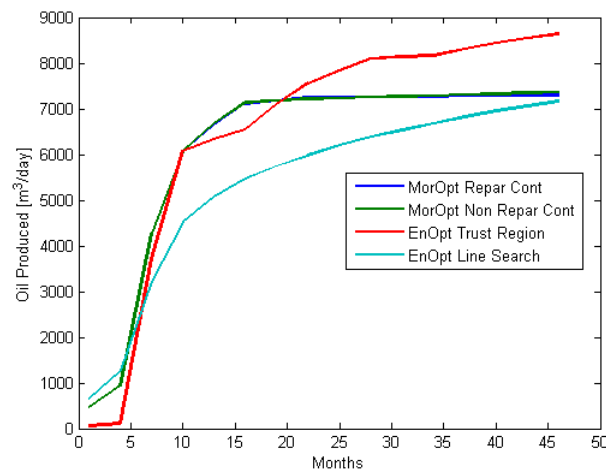


Figure 6-18: Oil Production Under Optimized Controls

Further analysis of Figure (6-17) reveals the water breakthrough times for the particular optimization approaches. It is easy to see that using optimized controls from EnOpt trust region delays the the water breakthrough time to nine months after the injection process has started. This is three months later that the water breakthrough time observed when using MorOpt. For a more detailed description of what is going on in Figures (6-16),(6-17) and (6-18) see Tabel (6-6).

from Table (6-6) a detailed picture emerges when trying to understand why the EnOpt line search optimized controls leads to such a low final NPV value. Even though the optimized controls present the highest volume of water injected and also the highest volume of water produced, not the same is true for the volume of oil

Case	Optimization Method	Inj Water [m^3]	Water Prod [m^3]	Oil Prod [m^3]	NPV [\$]
1	MorOpt Repar Cont	$1.4275 \cdot 10^4$	$0.6978 \cdot 10^4$	$7.2966 \cdot 10^3$	$7.9937 \cdot 10^5$
2	MorOpt Non-Repar Cont	$1.4546 \cdot 10^4$	$0.7179 \cdot 10^4$	$7.3678 \cdot 10^3$	$8.0294 \cdot 10^5$
3	EnOpt Trust Region	$2.1600 \cdot 10^4$	$1.2961 \cdot 10^4$	$8.6393 \cdot 10^3$	$8.1718 \cdot 10^5$
4	EnOpt Line Search	$2.4309 \cdot 10^4$	$1.7145 \cdot 10^4$	$7.1646 \cdot 10^3$	$5.4475 \cdot 10^5$

Table 6-6: Comparison of Total Liquid Volumes for 3D case

produced, which is the lowest. Looking at Figure (6-12) it is clear that the reservoir model contains high permeability channels, and injecting high volumes of water through injectors that are close to these channels will results in a fast transport of the water to the producers which are also along these channels.

6-3-1 Conclusions

The model-order reduced production optimization method was used to construct a optimized production strategy that involves water injection rates. It was done using a two-phase, (water-oil) reservoir simulation model that contains 18553 active cells. An ensemble of 25 members was generated in order to obtain the necessary snapshots.

- The results presented in this section contain information about four different optimization approaches, MorOpt with and without reparametrized controls, EnOpt trust region and EnOpt line search. These results suggest that the model-reduced approach provides results which are comparable with results coming from EnOpt. In terms of the final NPV that the optimized controls result in, three of the four approaches providing similar results. EnOpt line search gets stuck in a local maximum during the optimization process. Even though we use a fixed number of iteration for EnOpt method, the stochastic gradient constructed using just 25 ensemble members does not appear to help the method escape the local optimum it reaches.
- In terms of computational cost, MorOpt with reparametrized controls is better than the other three approaches, it is 38.45% less computational demanding than MorOpt without reparametrized controls, it is 33.96% less computational demanding than EnOpt using trust region.
- The reduction in the computation cost of MorOpt using reparametrized controls comes from the actual reparametrization of the control space. This results in fewer control patterns used in the construction of the reduced order model. Analysing MorOpt without reparametrized controls we observe that in order to construct the reduced-order model we need to run the full order model p times, where p is the number of saturation patterns, plus an additional 64 times (64 control elements in the control vector). The number of patterns that are obtained after POD depends on the total relative importance defined. For a 70% relative importance we expect around 9 saturation patterns and when it is increased to 95% we expect around 31 patterns.

- Using reparametrized controls we construct reduced-order model using the saturation patterns and the control patterns which for a relative importance of 95% results in 50 full order model runs. Which is significantly less than the 95 full order model runs needed for the construction of the reduced-order model while using MorOpt without reparametrized controls.
- The results from this section prove that MorOpt can be considered a viable optimization method. Also, choosing MorOpt as the preferred optimization method, most of the time will provide optimized controls which are not significantly lower than the results obtained using EnOpt trust region. On the other hand choosing MorOpt with reparametrized controls will results in an average reduction of the computational cost of 36% when compared with EnOpt or MorOpt without reparametrized controls.

6-4 Summary

In this chapter numerical results from the model-reduced gradient-based production optimization, based on the POD method were presented. The POD technique is used to identify a low dimensional subspace of the state space and control space, in which the dynamic of the reservoir states are well represented. Using this method a simplified linear form of the reservoir flow equations is derived and used to maximize an objective function. In both the 2D and 3D presented cases the model reduced approaches performed similar to the ensemble based optimization approach. This proves that MorOpt can be considered a viable optimization method.

Chapter 7

Conclusions

Two different models were used for the reservoir simulation experiments, a small 2D reservoir model and a relatively complex 3D reservoir model. Both models generate state variables of the order $10^3 - 10^5$. The aim of this thesis was to solve production optimization problems assuming that the simulator does not contain automatic gradient calculation for the desired controls and that the user does not have access to the simulation source code. The method studied, satisfying these assumptions is based on reduced-order modelling techniques, mainly the POD technique. The studied approach avoids implementing the adjoint method by replacing the original high order model with a linear low order approximation. The Jacobian involved in the construction of the low order model is approximated using finite differences. For the linearised low order model, an adjoint model is created by the transpose of the Jacobian. The most important parameters for this particular problem are the control values. They describe how much water is injected in the reservoir and through which injection well. Changing the injection rate results in a limited control of the subsurface flow. The ideal optimized control strategy is represented by the least amount of water injected that results in the maximization of the volume of produced oil and the minimization of the volume of produced water. The research objective of this thesis was to evaluate the performance a gradient based production optimization routine which involves a model-order reduction approach. Two different approaches were studied and compared with the ensemble optimization method, which is a different gradient based optimization technique. This chapter presents the conclusions of this thesis and recommendation for further research.

7-1 Conclusions

- The sparse structure of the partial differential equations is lost during the process of projection onto the reduce-order subspace. Leading to the conclusion

that the dense structure of the reduced order model does not imply extraordinary reduction in the computational time. This is reflected in the 3D study case results.

- Two synthetic reservoir models, that varied in complexity and size were used. In both cases the model reduced approach performed well. For the more simplistic 2D reservoir model the simulation results show that the method is not particularly sensitive to the background control vector. This shows that the method does not possess a particular tendency to stop at local optimums. The quality of the optimization in terms of highest NPV value was comparable to the quality of the optimization obtained by the ensemble based method. The examples considered show that implementing the model-order reduction optimization method reduces the overall computational cost with an average of 35% when compared with the ensemble based optimization method.
- The ensemble optimization approach, uses the statistical information from an ensemble of random perturbations of control variables, in order to estimate the cross-covariance between the objective function and the control variables. The cross-covariance contains information regarding the correlation between the control variables and the objective function.
- The ensemble optimization approach, is adjoint free and is relatively easy to implement around any reservoir simulator, without requiring access to the simulator source code. The POD-based model reduced method, also, does not require access to the simulator source code and is easy to implement. Both methods use a generated ensemble of controls. EnOpt uses the ensemble in order to construct the stochastic gradient, as the cross-covariance between the objective function and the controls, MorOpt uses the ensemble in order to construct the snapshots from which the dominant patterns required for the reduced order model are picked.
- In case of highly non-linear systems, the system matrices, $f_i(\Phi z(t_{i-1}), \theta)$ need to be recomputed at every time step and projected on the low order subspace, which is computationally expensive. Therefore, the POD method is more effective for linear cases, where system matrices have to be projected only once in order to capture the main dynamics of the system.
- In the model-reduced approach the approximated function is quadratic, and therefore the optimization problem solved using the reduced-order model is more simple than the optimization problem in terms of the original objective function. The model reduced gradient can be interpreted as an ensemble gradient which is supposed to prevent the method from converging to a local optima in case of a non-smooth objective function.
- The numerical experiments show that using only a few control and state patterns, the model reduced method provides approximate solutions for production optimization problems and relatively good approximations for the gradient. When very accurate gradients are required, POD-based model-reduced

optimization method may require high number of patterns. This automatically translates to a high computational cost.

- The initial approach proposed in this work has a few limitations. Since the sensitivity of the system with respect to the controls has to be approximated using finite-differences. Meaning that the perturbations needed in the model-reduced method plus the number of control parameters reaches a order larger than 100 and turns the method into a computationally demanding approach. The thesis proposes a way to avoid this limitation by implementing a variation to the MorOpt method, which involves reparametrizing not only the state space but also the control space. This implies using the POD method in order to derive not only state patterns, but also significant control patterns from the randomly generated ensemble of controls.

7-2 Further Research and Recommendations

- This thesis main focus was to investigate the application of the model-order reduction method for production optimization. The model-order reduction method used, incorporated the POD technique, which presents advantages for linear cases but does not offer the same advantages for non-linear cases. Further research can be done in incorporating different reduction techniques that perform better for non-linear system such as TPWL (Trajectory Piecewise Linearisation), which represent the non-linear system with a piecewise-linear system and then reduce each of the pieces with linear model reduction methods.
- Optimality of a particular basis does not imply optimality of the reduced order dynamical model. The method is highly dependent on the given data. Therefore, it is essential to choose snapshots that represent the dynamics of the model in the range of the parameters or control variables that we are interested in. However, there is no guarantee that the reduced order model will follow the same dynamics as the high order model.
- During the numerical experiments on the complex 3D model, the number of POD base functions preserving a certain fixed level of relative importance decreases during optimization. For this reason it is advised to use a large relative importance of 95-99% when building the reduced state subspace.

Appendix A

Appendix

A-1 Figure Appendix Section

This part contains figures from numerical experiments without extensive analysis.

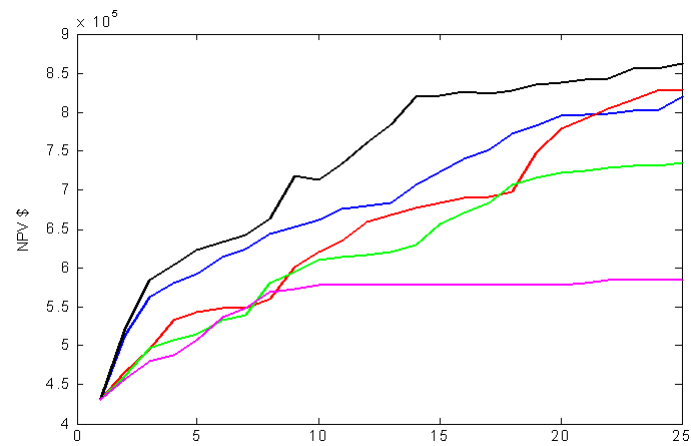


Figure A-1: NPV evolution: MorOpt reparametrize controls # Samples 25 relative importance 95% (blue line) MorOpt reparametrize controls # Samples 50 relative importance 95% (red line) MorOpt reparametrize controls # Samples 50 relative importance 75% (gree line) EnOpt # Samples 50 trust region (black line) EnOpt # Samples 50 line search (magenta line)

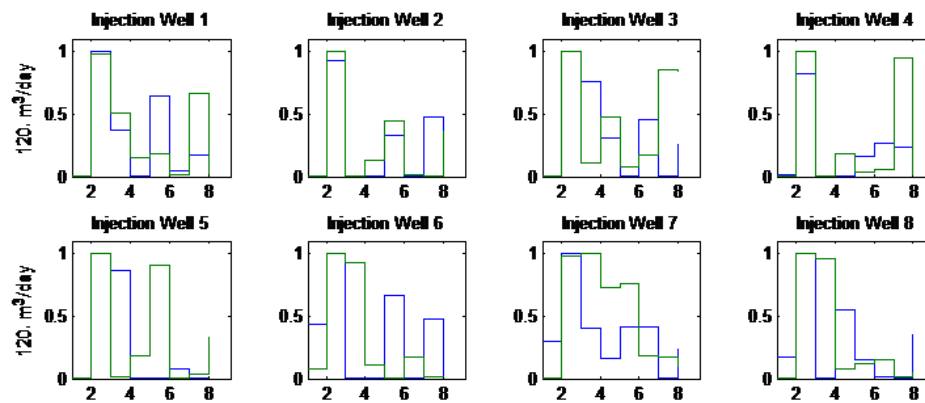


Figure A-2: Production strategy: MorOpt reparametrize controls # Samples 25 relative importance 95% (green line) EnOpt # Samples 50 trust region (blue line)

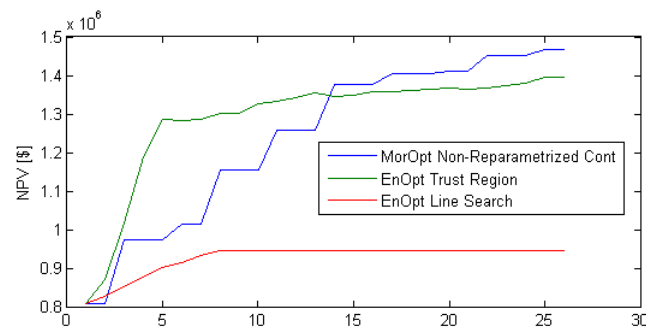


Figure A-3: NPV evolution

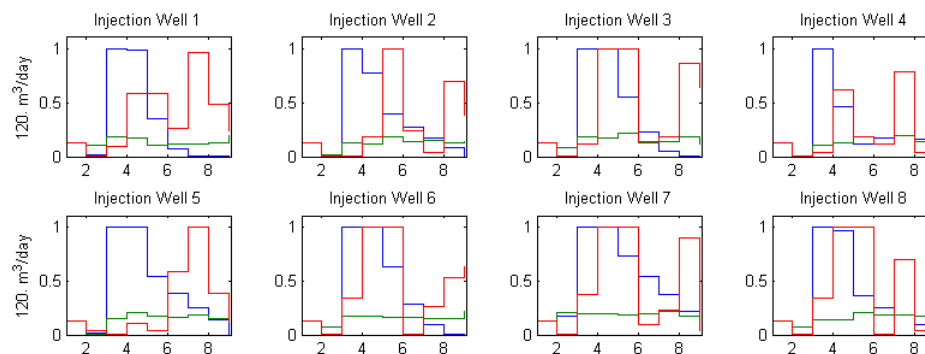


Figure A-4: Production strategy Figure (A-3)

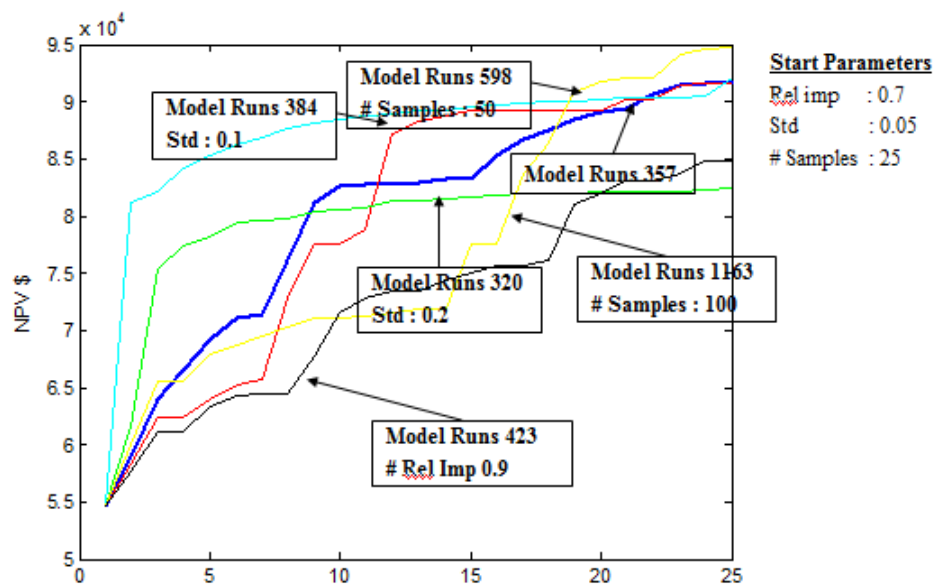


Figure A-5: 2D MorOpt Parameter Sensitivity Analysis

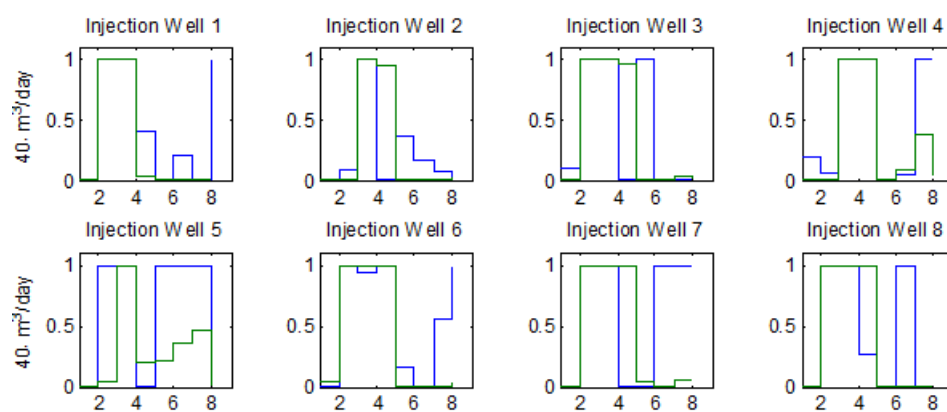


Figure A-6: 3D Production Strategy Comparison

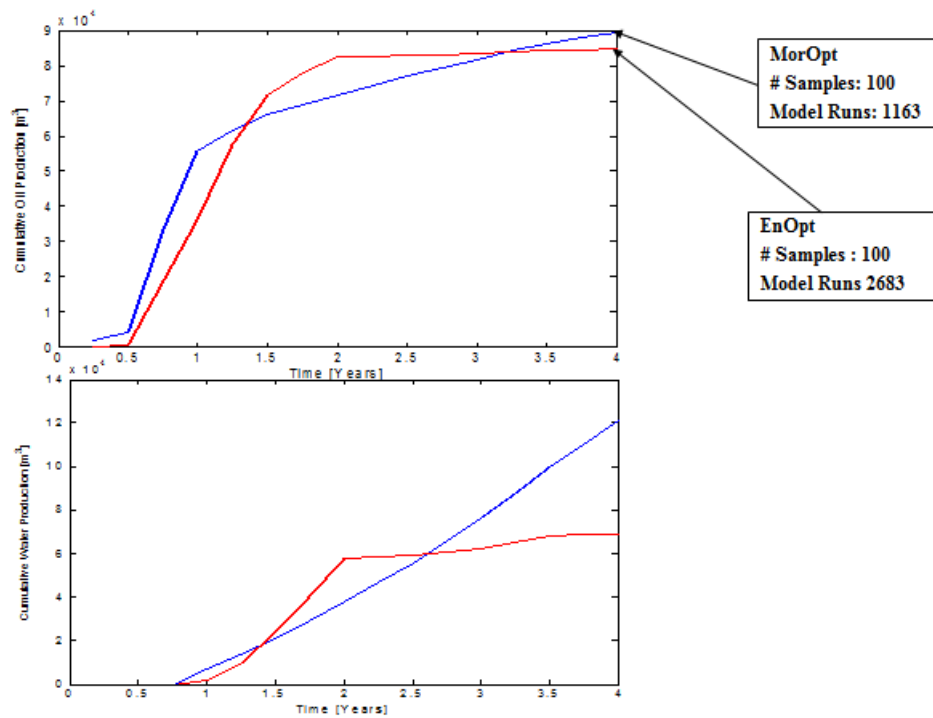


Figure A-7: 2D Cumulative Oil and Water Production

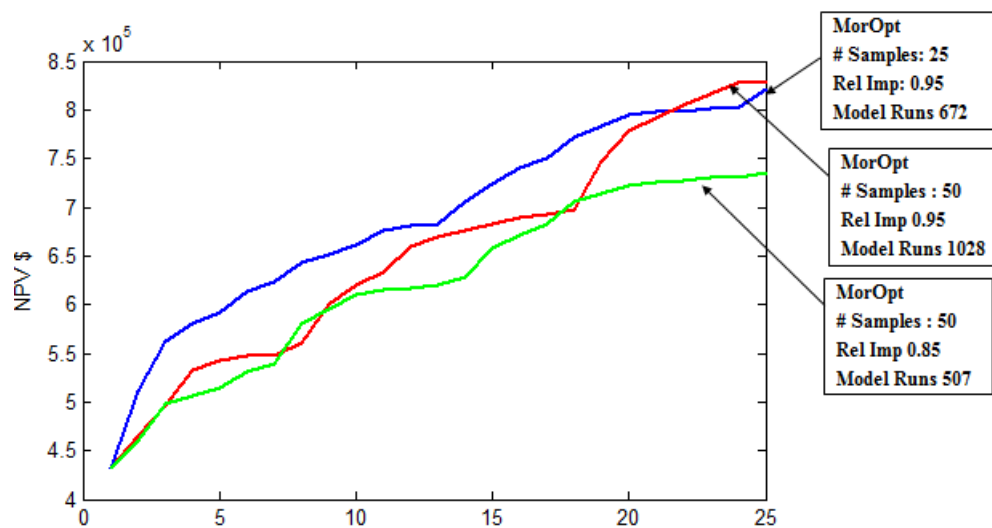


Figure A-8: 3D MorOpt NPV Comparison

Bibliography

- [1] Y. Chen, Dean S. Oliver *Efficient Ensemble-Based Closed-Loop Production Optimization*, Society of Petroleum Engineers (2008),SPE/DOE Improved Oil recovery Symposium Tulsa, Oklahoma, U.S.A. Apr 2008.
- [2] Y. Chen, Dean S. Oliver *Ensemble-Based Closed-Loop Optimization Applied to Brugge Field*, Society of Petroleum Engineers (2010),SPE Reservoir Simulation Symposium The Woodlands, Texas, U.S.A. Feb 2010.
- [3] Ho-Jeen Su, Dean S. Oliver *Smart Well Production Optimization Using an Ensemble-Based Method*, Society of Petroleum Engineers (2010),SPE Technical Symposium Al-Khobar, Saudi Arabia, May 2009.
- [4] O. Leeuwenburgh, Paul J.P. Egberts *Ensemble Methods for Reservoir Life-Cycle Optimization and Well Placement*, Society of Petroleum Engineers (2010),SPE/DGS Annual Technical Symposium Al-Khobar, Saudi Arabia, Apr 2010.
- [5] M. Asadollahi, G. Naevdal *Ensemble Methods for Reservoir Life-Cycle Optimization and Well Placement*, Society of Petroleum Engineers (2010),SPE/EAGE Reservoir Characterization and Simulation Conference Abu-Dhabi, UAE, Oct 2009.
- [6] M.P. Kaleta *Model-Reduced Gradient-Based History Matching*, TNO/ISAPP (2011),Doctoral Thesis TU Delft 2011.
- [7] O. Leeuwenburgh, Paul J.P. Egberts *Robust Life Cycle Optimization Work Flows*, Reference Document(2011),TNO ,Utrecht, Netherlands Dec 2011.
- [8] K.A. Lie, S.Korgstrad, I.S. Ligaarden J.R.Natvig, H.M. Nilsen, B. Skaflestad *Open Source MATLAB Implementation of Consistent Discretisations on Complex Grids*, SINTEF Applied Mathematics, Trondheim, Norway 2011.
- [9] Dean S. Oliver, Albert C. Reynolds, N. Liu *Inverse Theory for Petroleum Reservoir Characterization and History Matching* , Cambridge Univ. Press, Cambridge, 2008,

- [10] Jorn F.M. van Doren R. Markovinic, Jan-Dirk Jansen, *Reduced-order Optimal Control of Water Flooding using Proper Orthogonal Decomposition*, Computational Geosciences, 10:137-158, Springer 2006.
- [11] E. Garcia *SVD and LSI Tutorial 1: Understanding SVD and LSI*, <http://www.miilista.com/information-retrieval-tutorials/svd-lsi-1-understanding/>, Last update July 2007.
- [12] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee and C.G. Wu *Proper Orthogonal Decomposition and its Applications*, Journal of Sound and Vibration, 252(3):527-544, Elsevier Science Ltd. 2002.
- [13] M.A. Cardoso, *Reduced-Order Models for Reservoir Simulation*, Society of Petroleum Engineers (2009), SPE International Student Paper, SPE-129636-STU, New Orleans, U.S.A. Oct 2009.
- [14] M. Reza, M. Khaninezhad, *Hybrid Parametrization of Reservoir Properties for Robust History Matching under Geologic Uncertainty*, Society of Petroleum Engineers (2011), SPE Reservoir Characterisation and Simulation Conference, SPE-146934, Abu Dhabi, U.A.E . Oct 2011.
- [15] Anindya Chatterjee, *An Introduction to the Proper Orthogonal Decomposition*, Department of Engineering Science and Mathematics, Penn State University, Current Science vol 78, No7, Pennsylvania, USA, Apr 2000.
- [16] Lindsay I. Smith, *A Tutorial on Principal Component Analysis*, Feb 2002.
- [17] Wil Schilders, *Introduction to Model Order Reduction*, Eindhoven University of Technology, Faculty of Mathematics and Computer Science, Eindhoven, The Netherlands.
- [18] G. H. Golub and C. F. Van Loan. *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.
- [19] de Villemagne C., R.E. Skelton, *Model Reductions using a Projection Formulation*, International Journal of Control 46(6), 2141-2169, 1987.
- [20] Patricia Astrid, *Reduction of Process Simulation Models: a proper orthogonal decomposition approach*, Eindhoven : Technische Universiteit Eindhoven, PhD-study of the Dutch Programme EET, 2004.
- [21] Frank Vanden Berghen *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions* Universite Libre de Bruxelles Annee academique, Faculte des Sciences Appliquees 2004.
- [22] Michal Jerzy Rewienski, *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*, Phd-study, Technical University of Gdansk, Poland, 1998.
- [23] J.L. Lumley, *The structure of inhomogeneous turbulence*, Atmospheric turbulence and radio wave propagation, Nauka, Moscow, 166-178, 1967.

-
- [24] D.D. Kosambi, *Statistics in function space*, J. Indian Math. Soc. 7, 76-88, 1943.
 - [25] V.S. Pougachev , *General theory of random function*, Izv. Akad. Nauk. Set. Mat. 17, 1401-2 1953.
 - [26] A.M. Obukbov, *Statistical description of continuous fields*, Tr. Geophysics Int. Akad. Nauk. SSSR, 3-42, 1954.
 - [27] A. Papoulis , *Probability, Random Variables and Stochastic Process*, New York, U.S. 1965.
 - [28] A. Rosenfeld and A.C. Kak , *Digital Picture*, New York, U.S. 1982.
 - [29] R. Heijn, R. Markovinovic and J.D. Jansen, *Generation of low order reservoir models using system-theoretical concepts*, SPE Journal 9(2), 202-218, 2004.
 - [30] R. Markovinovic, E.L. Geurtsen, T. Heijn, and J.D. Jansen, *Generation of low order reservoir models using POD, empirical Gramians and subspace identification*, 8th European Conference on Mathematics of Oil Recovery, Freiberg, 2002.
 - [31] R. Markovinovic and J.D. Jansen, *Accelerating iterative solution methods using reduced-order models as solution predictors*, International Journal for Numerical Methods in Engineering, 68(5), 525-541 2006.
 - [32] M.A. Cardoso, L. Durlofsky and P. Sarma *Development and application of reduced order modelling procedures for subsurface flow simulation*, International Journal for Numerical Methods in Engineering, 77(9), 1322-1350 2009.
 - [33] Jorn F.M. van Doren, R. Markovinovic, Jan-Dirk Jansen, *Reduced-order optimal control of waterflooding using POD*, Proc. 9th European Conference on Mathematics in Oil Recovery (ECMOR IX), Cannes, France, 2004.
 - [34] C.A. Antoulas, *Approximating for large-scale Dynamical Systems*, Philadelphia, U.S, SIAM, 2005.
 - [35] B. Jafarpour and D.B. McLaughlin, *Reservoir characterization with the discrete cosine parametrization*, SPE Journal 14(1), 182-188, 2009.
 - [36] I. Shani and R.N. Horne, *Multiresolution wavelet analysis for improved reservoir description*, SPE Reservoir Evaluation and Engineering 2, 326-339.

